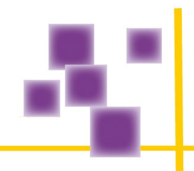




Livre Blanc

Intégration et Publication de logiciels sous licences Open Source : Mode d'emploi

Aspects juridiques, économiques et communautaires

Version 1.0 - Décembre 2010



RÉSUMÉ

Au sein de Telecom Valley, la Commission Open Source constituée en 2007, regroupant différents acteurs du domaine, a constaté que de nombreuses entreprises, quelle qu'en soit la taille, installées en Région Provence Alpes Côte d'Azur mais aussi, sur le Territoire National font face à des interrogations récurrentes concernant le logiciel dit Libre :

- Quels sont les différents modèles économiques du Libre ?
- Quelle est l'origine et quels sont les vecteurs de la création de valeur ?
- Quelle approche juridique concernant les licences Open Source ? et existe-t-il des risques associés ?
- Comment créer et animer une communauté « Open Source » ?
- Est-il nécessaire de mieux communiquer autour de l'utilisation du Logiciel dit Libre ?
- Comment avoir une plus grande clarté vis-à-vis du marché ?

Nos premiers travaux nous ont amené à rédiger ce Livre Blanc présentant les principes fondamentaux des Logiciels Libres :

- Récapitulant un inventaire des modèles économiques,
- Incluant une analyse juridique des licences,
- Incluant une analyse économique relative à la création de la valeur et aux enjeux.

Ouvrage collectif, il repose sur l'expérience cumulée de l'ensemble des membres de Telecom Valley, des acteurs majeurs de l'Open Source et des contributeurs bénévoles de notre Commission Open Source.

Ce Livre Blanc s'adresse aux éditeurs de logiciel, aux décideurs informatiques des entreprises des secteurs public et privé, aux collectivités publiques, aux dirigeants politiques qui souhaitent mieux comprendre l'économie et les enjeux du Logiciel Libre, aux investisseurs, aux journalistes en quête perpétuelle d'informations et d'exemples pour illustrer l'un des secteurs les plus dynamiques de l'innovation logicielle.
Ce document est plus largement destiné à toutes les personnes qui s'intéressent à l'émergence des Logiciels Libres dans le monde entier.

Il faut bien comprendre que le Logiciel Libre n'est pas une nouvelle technologie. Il s'agit avant tout d'un mode de production et de distribution du logiciel, c'est-à-dire un ensemble de pratiques concernant le management des projets, la gestion du code source, les canaux de mise à disposition des produits, le régime légal d'utilisation et de commercialisation, l'assistance technique, et même le marketing et les forces de vente.

Ce mode de production n'est pas en soi une innovation. La notion de Logiciel Libre a été formalisée dès les années 1980, et les pratiques correspondantes sont sans doute encore plus anciennes. On peut même dire que le Logiciel Libre est né dès que la notion de « produit logiciel » indépendant du matériel a été perçue par les développeurs et les utilisateurs. Son apparition a silencieusement précédé celle de l'édition commerciale de logiciels non libres telle que nous la connaissons aujourd'hui. L'innovation réside dans l'ampleur et dans les usages du phénomène « Logiciel Libre » qui se manifeste sur plusieurs plans.

Naguère réservé à des initiés, le Logiciel Libre est devenu, avec la généralisation de l'Internet, l'un des outils structurant ce « village global » propice aux modes de développement collaboratifs et décentralisés. La visibilité des projets libres en a été immédiatement amplifiée.

Les Logiciels Libres se sont invités dans tous les segments de l'édition logicielle, y compris les applicatifs « métiers » (progiciels de gestion intégrés, gestion de la relation client, business intelligence) et la bureautique. Ils touchent ainsi les utilisateurs et les décideurs.

Un écosystème complet est, à présent, en place et, de la production à la mise en œuvre chez l'utilisateur, chacun peut y trouver ce qui correspond à son métier et à ses objectifs. Le Logiciel Libre n'est plus seulement une matière première pour informaticiens et, sur un modèle où l'accès aux sources est par définition ouvert, il existe des produits finis et des intégrateurs qui s'engagent sur des résultats.

La question de l'appropriation des aspects juridiques des Logiciels Libres constitue un élément majeur du Business model pour les entreprises et les collectivités publiques. Le Logiciel Libre peut être librement exécuté, copié, distribué, étudié, modifié et amélioré, sous réserve de respecter - le plus souvent - une condition essentielle : la rediffusion du code source modifié ou non, doit, elle-même respecter ces quatre libertés. Si cette condition n'est pas respectée, on considère qu'il y a contrefaçon du fait du non respect de la licence d'origine. Le choix de la licence est déterminant et ses conséquences juridiques doivent être dûment envisagées. Il est vrai qu'il n'y a, aujourd'hui, que peu de contentieux. Pourtant, les auteurs de logiciels libres n'hésitent plus à porter devant les tribunaux tout manquement contractuel. La jurisprudence applicable en la matière est en train de se dessiner.

Les raisons de l'extraordinaire expansion du modèle libre dans la présente décennie sont diverses.

L'accès souvent gratuit au droit d'usage, sans être la caractéristique essentielle du Logiciel Libre, exerce un pouvoir d'attraction à court terme mais son influence sur le bilan économique global des projets est cependant limitée et ne saurait être le fondement d'une réussite durable.

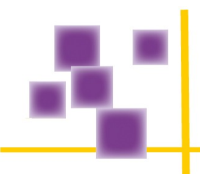
Le choix de recourir à un Logiciel Libre ne doit pas être fait par défaut mais en connaissance de cause, en maîtrisant chacun des aspects (technique, juridique, organisationnel et commercial) du choix ainsi fait.

Les limites de l'édition logicielle traditionnelle attachée au « secret de fabrique » (logiciel propriétaire) sont maintenant perçues par un public de plus en plus large. La seule « *assurance-sérénité* » pour l'utilisateur réside-t-elle désormais dans le service associé au logiciel, lui-même développé sur le principe de la « *source ouverte* » ?

L'émergence des modèles basés sur le Logiciel Libre ou plus récemment le « logiciel en tant que service » (Software As A Service - SAAS) aussi bien dans le monde libre que le monde propriétaire remettent en cause le modèle de rémunération à la copie. Assiste-t-on à une nouvelle étape commerciale de l'histoire du logiciel ?

Enfin, les coûts structurels de distribution et de protection des logiciels propriétaires peuvent rendre beaucoup plus chères la production et la diffusion de logiciels. La capitalisation sur le code en est de plus amenuisée puisque réduite à une seule entreprise : la productivité dans le secteur du logiciel propriétaire semble marquée par son absence d'efficacité. Un recadrage du marché est là encore en mouvement.

L'objet de ce Livre Blanc est de présenter une synthèse des pratiques, des enjeux et des opportunités du Logiciel Libre, dont la réussite soulève encore des interrogations économiques. Tout porte à croire qu'aujourd'hui l'environnement est favorable et l'essentiel des savoir-faire est en place, la situation économique présente (et son corollaire, la réduction drastique des budgets) amène beaucoup





d'éditeurs et d'entreprises utilisatrices à se poser une question inenvisageable il y a encore quelques mois : Pourquoi ne pas éditer mon logiciel sous licence libre ? Pourquoi ne pas intégrer des Logiciels Libres dans les développements internes et dans mon système d'information ? Comment en vivre ?

Nous avons décidé d'organiser le travail de nos contributeurs de la façon suivante :

Dans un premier temps, nous avons recensé **les idées reçues, les objections les plus courantes et les "fausses barbes"**, utilisées, souvent de bonne foi, pour justifier l'impossibilité d'envisager l'édition sous licence Open Source d'un logiciel ou son utilisation dans le système d'information de l'entreprise.

Nous avons ensuite présenté un **petit historique de l'édition logicielle** dans lequel nous montrons que les Logiciels Libres ne sont pas forcément une tendance récente de l'industrie des Technologies de l'Information et de la Communication et que les modèles propriétaires/libres se sont développés concomitamment pendant de nombreuses années.

Puis nous avons abordé **l'aspect de l'écosystème et des business model**, pour tenter de répondre aux questions récurrentes qui nous sont posées : Est-il possible de gagner de l'argent avec les Logiciels Libres ? Existe-t-il un ou plusieurs business models ? Lequel choisir pour maximiser mes chances de succès ?

Le précédent chapitre nous a logiquement amené à aborder les questions relatives au **régime juridique des Licences Open Source**, les obligations en découlant, la responsabilité attenante, et certains points particuliers (Copyleft, etc.).

L'avant dernière partie de ce Livre Blanc, nous a amené à recenser **les leviers que procurent le choix d'une licence libre** pour l'éditeur ou d'un logiciel pour un directeur des systèmes d'informations : quels sont les bénéfices tangibles, immédiats et plus long terme, financiers, marketing, juridiques qui découlent des logiciels libres.

Enfin, en guise de conclusion, nous exposons les **limites respectives des différents modèles, Open Source et Propriétaire**.

Les terminologies « Logiciel Libre » et « Open Source » présentent quelques différences, essentielles pour certains, anodines pour d'autres. On pourrait se contenter de l'expression FLOSS (Free/Libre Open Source Software) qui qualifie à la fois les Logiciels Libres et les logiciels Open Source, les deux grands mouvements soutenus respectivement par la FSF[1] et l'OSI[2]. Dans ce Livre Blanc nous avons essayé d'avoir une approche pragmatique. Malgré les différences existantes entre ces deux termes, au risque de nous attirer les foudres de puristes de tous bords, nous avons décidé d'utiliser indifféremment l'un ou l'autre.

La notion d'Open Source constitue désormais un mode alternatif de conception industrielle.

Librement vôtre,


[1] FSF : Free Software Foundation


[2] OSI : Open Source Initiative



Les membres contributeurs du groupe de travail Open Source de Telecom Valley :

*Pascal AGOSTI, Avocat associé, Docteur en Droit, Cabinet Caprioli & Associés, http://caprioli-avocats.com

*Marc AUGIER, Professeur de management des systèmes d'information, SKEMA, http://www.skema.edu

*Jean DEMARTINI, Fondateur dirigeant DEMTECH, http://www.demtech.net

*Christophe DENEUX, Integration Architect, Capgemini Sud, http://www.fr.capgemini.com ; contributeur au projet Petals ESB http://petals.ow2.org

*Pascal FLAMAND, Fondateur-Dirigeant Janua et Responsable de la Commission Open Source de Telecom Valley, http://www.janua.fr

*Juliette MALLEZ, Consultante KM et Communication, @CTIS-Ingénierie http://www.actis-ingenierie.com

Relecteur et contradicteur :

*Jean-Christophe BECQUET, Fondateur-Dirigeant d'APITUX et Vice-président de l'April, http://www.apitux.com/

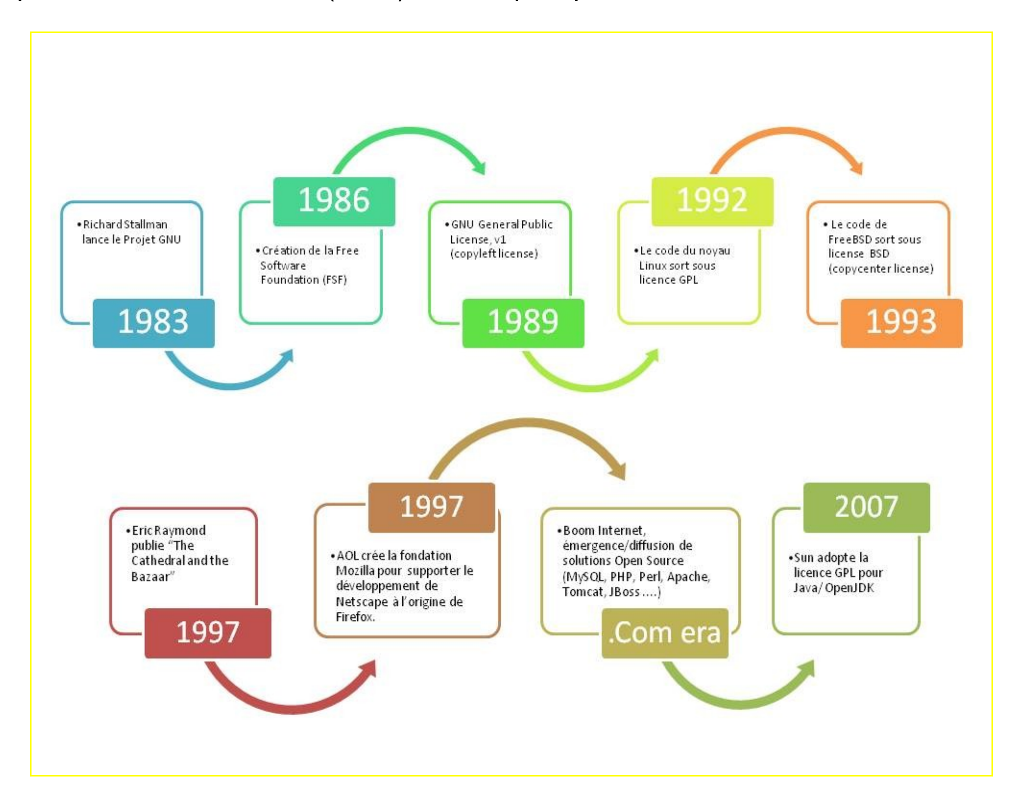
*Patrice BERTRAND, Directeur Général Smile et Président du Conseil National du Logiciel Libre, http://www.smile.fr

*Jean Christophe PAZZAGLIA, Vice Directeur Recherche - SAP-Labs France, http://www.sap.com

TABLES DES MATIÈRES

1. INTRODUCTION

Les logiciels libres existent officiellement depuis 1985, année de la création de la Free Software Foundation par Richard M. Stallman (RMS) donc depuis près de 25 ans.



Historique de l'Open Source

En Français, nous avons la chance de ne pas avoir d'ambiguïté dans le terme: le mot "libre" fait référence à la liberté et non pas au prix. La définition exacte du logiciel libre est donnée par la Free Software Foundation, en particulier sur son site.

L'expression « Logiciel libre » fait référence à la liberté pour les utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel. Plus précisément, elle fait référence à quatre types de liberté :

- La liberté d'exécuter le programme, pour tous les usages (liberté 1) ;
- La liberté d'étudier le fonctionnement du programme, et de l'adapter à vos besoins (liberté 2). Pour cela, l'accès au code source est une condition requise ;
- La liberté de redistribuer des copies,(liberté 3) ;
- La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté (liberté 4). Pour cela, l'accès au code source est une condition requise.

Un programme est un Logiciel Libre si les utilisateurs ont toutes ces libertés. Ainsi, vous êtes libre de redistribuer des copies, avec ou sans modification, gratuitement ou non, à tout le monde, et partout. Être libre de faire ceci signifie (entre autres) que vous n'avez pas à demander ou à payer pour en avoir la permission.

En revanche, le terme « Open Source » dérive d'une définition de l'Open Source Initiative (OSI), fondée par Eric Raymond et d'autres passionnés en 1998, dans le but de proposer une alternative aux Logiciels Libres qui respecte l'idée initiale de partage en proposant un modèle plus en adéquation avec les interrogations des entreprises. Les valeurs du modèle Open Source sont plutôt les valeurs de l'ingénieur, principalement pratiques, qui cherchent une façon plus efficace de produire du code.

Selon la définition de l'OSI, la licence open source doit répondre à cette série de critères pour être qualifiée en tant que telle :

- la libre redistribution du logiciel ;
- le code source doit être fourni et accessible ;
- l'autorisation d'œuvres modifiées ou dérivées sous les mêmes conditions que la licence d'origine ;
- l'intégrité du code doit être préservée ;
- pas de discrimination entre les personnes ou les groupes ;
- pas de discrimination entre les domaines d'application ;
- la licence est indépendante des autres contrats ;
- la licence ne doit pas être propre à un produit ;
- la licence a un logiciel ne doit pas s'étendre à un autre ;
- la licence doit être neutre technologiquement.

1.1. Quelques rappels historiques

Un petit retour en arrière va nous permettre de mieux comprendre comment se sont créées les fondations à l'origine des mouvements connus aujourd'hui sur les vocables de Logiciels Libres ou Open Source. Nous verrons aussi que si nous avons l'habitude de séparer la population des développeurs logiciels en deux, et de les opposer : ceux qui écrivent des Logiciels Libres et ceux qui écrivent des logiciels propriétaires, la réalité est en fait plus complexe et de ce fait, la compétition entre ces deux communautés ne devrait pas prendre autant d'importance. De toutes les façons il ne faut pas perdre de vue que la majorité du code source produit dans le monde est écrit par des sociétés de service pour leurs clients, sans licence explicite.

En mettant à disposition le code source des logiciels qu'ils écrivent, et en offrant le droit à utiliser, modifier et distribuer ces modifications, les développeurs de Logiciels Libres ont bouleversé les cultures et les pratiques dans le monde du développement. Ce mode de développement, maintenant reconnu et officialisé, a donné ses lettres de noblesse à un mode plus proche du travail des artisans et des compagnons. De plus, il a permis de dynamiser l'industrie du logiciel et de la rendre plus efficace. Cette dernière affirmation est peut-être à tempérer, il est vrai qu'elle provient plutôt de gens favorables aux Logiciels Libres. Force est toutefois de constater que dans des domaines comme le Web les technologies issues de projets libres sont omniprésentes : Apache pour les serveurs Web, PHP comme langage de script pour ne citer que les plus célèbres.

Mais, à y regarder de plus près, cette création ne marquait pas l'invention d'un nouveau mode de développement. C'était au contraire une manière de sauvegarder cette méthode face à l'arrivée des logiciels à code fermé ou « propriétaires ». En effet, quand RMS[3] a créé la Free Software Foundation[4] (FSF) en 1985 c'était en réaction à la disparition de l'accès au code source des logiciels qu'il utilisait. Le mode de distribution des logiciels était en train d'évoluer, ce n'était plus le code source qui était envoyé

[3] Pour rappel : Richard M. Stallman. Ses initiales sont aussi son surnom et il aime bien se faire appeler 'RMS' à l'intérieur du mouvement qu'il a fondé.

[4] http://www.fsf.org

au client (ce qui bien sûr lui permettait de le modifier). Au lieu de cela, le code binaire directement utilisable était fourni, en contrepartie rendant impossible les bricolages, le *hacking*.

La biographie de RMS[5] nous renseigne sur les circonstances de la création de la FSF. Parmi les évènements qui ont contribué se trouve le remplacement d'une imprimante qui devait apporter plus d'efficacité à son équipe mais qui s'était révélée une source de frustrations. Ceci parce que le logiciel de contrôle était devenu “à code fermé”, donc verrouillé par le constructeur de l'imprimante. Par conséquent, les modifications qui leur avaient permis de maximiser l'utilisation de l'ancien modèle ne pouvaient plus être faites et le nouveau modèle, bien que doté d'un matériel plus moderne et plus performant, se retrouvait beaucoup moins pratique à utiliser et plus lent. Le matériel avait évolué, le logiciel non et il n'y avait plus accès pour répondre à ses besoins. C'est en particulier la fermeture du code source d'Unix avec l'arrivée de UNIX System V, Release 1 en 1983, qui a conduit au lancement du projet GNU, un système compatible Unix qui ne serait pas Unix pour éviter les problèmes de licence, et du projet GNU à la FSF pour l'héberger et lui donner un cadre juridique.

1.2. Faut-il opposer Logiciels Libres et logiciels propriétaires ?

Oui, bien sûr ! Mais disons plutôt comparer et les conclusions vont dépendre des critères de comparaison choisis. Ainsi, on peut essayer de comparer :

- l'exacte adéquation au besoin (qualité),
- la pérénnité potentielle,
- les potentialités d'évolutions,
- les coûts de développement,
- les techniques-méthodes de développement,
- les modalités de distribution,
- les techniques de support,
- etc.

En fait, tous ces critères ne sont pas forcément « pertinents » selon le point de vue qu'on adopte. Aucun choix ou aucune décision ne s'impose à l'évidence. Les deux approches ont leurs raisons d'être. Aider à faire ce choix est, d'ailleurs, l'objectif de ce document.

Du point de vue de l'utilisateur, cette possibilité de choix est un facteur de liberté fondamental.

Pour être franc, les critères techniques sont assez vite peu pertinents dans la mesure où les développeurs impliqués dans le développement d'un logiciel distribué sous licence libre ou sous licence propriétaire sortent des mêmes écoles, ont lu les mêmes livres, pratiquent la même informatique bien qu’utilisant souvent des langages, des méthodologies et des frameworks différents.

Si la différence n'est pas technique, où réside-t-elle ?

Elle correspond pour certains à une différence de vision de la valeur de l'information qu'on a à manipuler et du pouvoir que sa possession assure :

- Vision 1 : L'information est tellement *importante* qu'il vaut mieux la **divulguer** plutôt que de la *perdre*. Cette vision correspond à la conception – Logiciel Libre.
- Vision 2 : L'information est tellement *précieuse* qu'il vaut mieux la *perdre*[6] que de la *divulguer*. Cette vision correspond à la conception – Logiciel Privateur[7].

[5] Williams, Sam, Free as in Freedom: Richard Stallman's Crusade for Free Software, *O'Reilly,* 2002

[6] Comme dans les histoires de pirates où le chef tue tous ceux qui l’ont aidé à enterrer le trésor.

[7] Expression imagée introduite par Richard Stallman pour nommer les logiciels propriétaires.

Selon le contexte telle ou telle vision peut être considérée comme pertinente.

On parle beaucoup, en ce moment et sans doute pour longtemps, de développement durable. Si notre humanité a duré et s'est développée n'est-ce pas à cause d'une transmission[8] systématique des informations et des connaissances ? Faut-il prendre le risque d'interrompre ce parcours en coupant cette transmission ?

On peut se demander pourquoi recourir à un Logiciel Privateur ?

Choisir et utiliser un logiciel entraîne un risque (comme pour tout outil technique). Choisir un Logiciel Libre, c'est accepter de l'assumer explicitement. Choisir un logiciel privateur c'est, comme croient certains, reporter ce risque à travers un contrat sur les épaules du fournisseur qui s'engage à obtenir des résultats ou à engager des moyens. Mais lorsqu'on examine les licences habituellement associées à un logiciel privateur, on se rend vite compte que ce report de risque est illusoire sur un plan pratique et juridique[9].

[8] Invention du livre, d'Internet et plus largement de la méthode scientifique basée sur la publication des travaux pour qu'ils puissent être validés et réutilisés par d'autres.

[9] Même si un très long procès peut être, éventuellement, gagné.

2. IDÉES REÇUES

Plusieurs idées reçues, bonnes et mauvaises / positives et négatives, circulent à propos des logiciels libres et des logiciels « propriétaires ». Ce chapitre se propose de les exposer et de les commenter afin de les transformer en affirmation ou de leur tordre définitivement le cou.

2.1. L'Open Source est gratuit

Il est souvent assimilé le terme « gratuit » au terme « logiciel libre ». Cette confusion provient de la traduction française du terme anglais « free software » qui signifie «logiciel libre » et non « logiciel gratuit ». Un logiciel libre est un logiciel pour lequel l'utilisateur a accès au code source avec certains droits s'y rapportant (redistribution, modification, …) déterminés par la licence du logiciel (GPL, LGPL, …).

En outre, il est vrai aussi que la majorité des logiciels libres disponibles sont accessibles gratuitement (certains ne le sont pas). La gratuité des logiciels libres n'est pas une obligation.

2.2. L'Open Source ne coûte rien

Nous venons de voir que la majorité des logiciels Open Source sont gratuits, mais est-ce pour autant qu'ils ne coûtent rien ? Comme n'importe quel logiciel, avant de le mettre en production, il faut s'assurer de la disponibilité d'un support, éventuellement associé à un engagement de qualité de service (Service Level Agreement), afin de régler les problèmes qui pourraient surgir en production, et éventuellement mettre à mal l'activité de l'entreprise.

Deux possibilités se présentent alors :

1. Le logiciel Open Source est édité par une société commerciale : dans ce cas il y a de grandes chances pour que cette société propose une prestation de support. Il suffit alors de la lui acheter,

2. Le logiciel Open Source n'est pas édité par une société commerciale : il existe des sociétés (en général SSII et SSLL) proposant des prestations de support de logiciels libres.

2.3. L'Open Source est de mauvaise qualité

La qualité d'un logiciel n'est pas liée au fait qu'il soit libre ou propriétaire mais à la qualité du travail effectué. Cependant l'accès au code source pourra faciliter la maintenance et la résolution des problèmes de fonctionnement. Avoir accès au code source n'est pas suffisant, encore faut-il être capable de l'analyser et le comprendre. C'est alors qu'entrent en jeu la communauté des développeurs, des contributeurs et les SSII/SSLL (au travers de forums, de mailing listes ou de contrats de support payants, vous pouvez poser vos questions, remonter vos bugs… et obtenir le plus souvent un accès direct et une réponse du/des développeur(s) du produit et une correction souvent plus rapide que dans le monde propriétaire) .

Le code Open Source n'est donc pas intrinsèquement de moins bonne ou de meilleure qualité qu'un code propriétaire, en revanche l'ouverture du code et le respect des standards donnent certaines garanties (pérennité, revue par les pairs, possibilité d'intervenir directement soi-même…).

2.4. L'Open Source est un monde d'idéalistes

Sans doute…L'ouverture du code, le partage, la contribution, la participation à l'élaboration et à la maintenance d'un logiciel, s'oppose au modèle du logiciel propriétaire porté par une seule entreprise.

Cette communauté est complétée depuis quelques années par l'apparition de modèles économiques viables portés par l'émergence d'éditeurs « libres » et de SS2L. Nous reviendrons plus tard dans ce livre blanc sur l'émergence de cet écosystème au chapitre 3.

2.5. L'Open Source est un outil pour spécialistes

Il est vrai que beaucoup de logiciels Open Source sont des logiciels middleware destinés à des serveurs :

- Système d'exploitation : Linux, BSD,…,
- Logiciels d'infrastructure : serveur web (Apache), serveurs de messagerie, …,
- Logiciels de sécurité, …,
- Bases de données, …,
- Serveurs d'application,… .

Mais il faut aussi reconnaitre qu'aujourd'hui :

- L'installation d'une distribution Linux est aussi simple que celle d'un OS propriétaire pour un utilisateur final.
- Les logiciels bureautiques sont maintenant de bonne qualité et utilisés par des millions d'utilisateurs (ex: OpenOffice).
- Des logiciels métiers sont largement reconnus, notamment dans le monde de l'ERP (Entreprise Resource Planning) avec par exemple Compiere (http://www.compiere.org/), du CRM (Customer Relation Management) comme SugarCRM (http://www.sugarcrm.com) ou Dolibarr (http://www.dolibarr.fr/) et de l'EDA (Electronic Design Automation) avec, par exemple, le projet gEDA (http://www.gpleda.org/).

Pour trouver des catalogues approfondis énumérant l'offre disponible en logiciels libres, il est possible de consulter sur Internet des sites tels que Framasoft (http://www.framasoft.net), l'annuaire FSF/UNESCO (http://directory.fsf.org/), etc. Le site http://www.click2try.com/ propose même de tester en ligne certains logiciels libres.

2.6. La disponibilité des sources ne sert à rien

Ce n'est pas parce que l'on n'est pas programmeur qu'avoir accès aux sources n'est pas intéressant. Prenez le cas suivant :

- Une société achète un logiciel propriétaire lambda. Elle l'utilise pendant plusieurs années sans aucun problème.
- Vient un jour où il lui est nécessaire de changer son parc de machines.

- Mais hélas pour elle, le logiciel ne fonctionne plus sur le nouveau système d'exploitation.
- Impossible pour elle de revenir au système d'exploitation antérieur.
- Il lui faut donc chercher une mise à jour de son logiciel.
- Mais, encore hélas, l'éditeur n'existe plus, et le logiciel non plus.
- Si elle avait eu accès aux sources (et que celles-ci avaient été libre de droit), elle aurait sûrement pu faire adapter son logiciel pour qu'il fonctionne sur ces nouvelles machines (seule ou avec l'aide d'une SS2L).

C'est pourquoi, même dans le cas de logiciels propriétaires, il est toujours important d'avoir accès aux sources, soit directement soit par dépôt auprès d'un organisme agréé, comme Logitas (http://www.logitas.com), l'Agence de Protection des Programmes (http://app.legalis.net) ou encore le site JeDepose.com (http://www.jedepose.com). La pérennité du logiciel en sera d'autant augmentée (et bien sur dans la mesure où l'aspect légal a été étudié).

2.7. Si je mets mon logiciel en Open Source, je vais me le faire voler

Une licence open source est une « vraie » licence qui donne des droits mais aussi des devoirs à l'utilisateur et protège fortement la propriété intellectuelle du créateur, et justement le protège contre toute utilisation indésirable.

De plus, comme les licences de Logiciels Libres imposent de fournir le code source et donnent des droits de modifications et distributions, certes sous certaines réserves, il est vrai que toute personne peut créer sa propre version (aussi appelée « fork »). Ceci est la théorie, il ne faut pas oublier ce qui se passe en réalité :

- La création d'un fork ne vous enlève pas la gestion de votre version,
- La création d'un fork demande un investissement conséquent de la part de son initiateur : sauf s'il est déjà un contributeur actif, l'initiateur du fork ne maitrisera pas le code source et toutes les subtilités. Il lui faudra aussi se faire connaître, ce qui n'est pas sans effort non plus.
- Les facteurs déclenchant la création d'un fork sont plutôt de l'ordre de l'humain : insatisfaction d'un membre ou d'un groupe de membres de la communauté avec le « project leader » sur, par exemple, des choix stratégiques ou techniques.
- La sélection sera naturelle, faite par le marché : c'est donc à vous de démontrer que votre version est meilleure que la version forkée et d'avoir toujours un coup d'avance par des mises à jour régulières, de la réactivité sur la correction des problèmes, de la documentation, une communauté active.

2.8. Aucune garantie n'est fournie avec un logiciel Open Source

Il est vrai que les logiciels Open Source n'offrent aucune garantie - dommages, pertes d'exploitation. Mais qu'en est-il réellement pour les logiciels propriétaires ?

De prime abord, on peut penser qu'une entreprise qui achète et utilise un logiciel commercial peut se retourner contre l'éditeur du logiciel si ce dernier est la cause d'une perte de business ou d'activité. Mais en regardant de plus près la licence du logiciel, on s'aperçoit qu'elle contient la plupart du temps une clause limitative ou d'exonération de responsabilité. La lecture des licences de logiciels est donc un prérequis essentiel pour déterminer l'étendue exacte des garanties offertes par l'éditeur et montrera rapidement que même pour les logiciels propriétaires les garanties sont très réduites, sans offrir en contrepartie la transparence et la liberté d'utilisation du libre.

D’autre part, se sont créées depuis quelques temps des sociétés de service et éditeurs proposant pour le monde Open Source des garanties similaires – support, correction de bugs, engagement de résultat… et prenant des engagements contractuels identiques aux modèles propriétaires.

3. LES LOGICIELS LIBRES, UN ÉCOSYSTÈME DIVERSIFIÉ EN OSMOSE AVEC INTERNET

La protection du code source peut sembler le meilleur moyen de protéger son capital. Pour une société développant des logiciels, cacher le code source c'est cacher ses recettes, ses secrets et donc protéger sa propriété intellectuelle. D'autre part, à partir du code source d'une application, il est facile d'en commencer une autre et donc pour un concurrent de développer une application concurrente, améliorée.

Tout ceci n'est, bien sûr, vrai que dans un monde où il est légal de masquer son code source, s'il était illégal de le faire, comme dans le monde protégé par la licence GPL, alors ce risque n'existerait plus : le concurrent devrait afficher à tous que son produit n'est qu'une évolution. D’autre part, pour être précis, la diffusion du code source n’est due qu’en cas de distribution et ce à l’intention du destinataire de la distribution.

Mais, cacher le code source nécessite de se débrouiller seul pour développer ses logiciels. A titre d’exemple, l’écriture d’un nouveau système d’exploitation requière des investissements colossaux que peu d’entreprises peuvent aujourd’hui se permettre (Microsoft et Vista en 2007).

En comparaison, Mandriva ou RedHat sont capables d'intégrer un système d'exploitation GNU/Linux avec une équipe de développement interne réduite. Leur travail est essentiellement du packaging : assembler les morceaux qui semblent les plus pertinents pour construire leur système,ou améliorer les utilitaires spécifiques tels que l'installation.

Alors que la principale raison de cacher le code source est de pérenniser l'investissement en propriété intellectuelle, nous nous rendons compte que c'est un jeu dangereux et que peu sont capable de rentabiliser.

Ce qui s'est passé avec Netscape en est une illustration[10]. Dans la fin des années 1990 Netscape et Microsoft étaient en compétition sur le marché des navigateurs Internet (browsers). Après des débuts impressionnants, Netscape a dû jeter l'éponge devant les ressources de Microsoft. La guerre des browsers semblait chose entendue, mais ce n'était en fait qu'une bataille et Netscape n'avait pas dit son dernier mot. Après avoir été rachetée par AOL Corporation, l’analyse de Netscape Communication est que pour continuer à exister il lui fallait s'appuyer sur un autre modèle que celui utilisé jusqu'à présent. En 1998, le code source du navigateur Netscape était confié à la communauté Open Source sous la forme d'une licence appelée la Mozilla Public License. Aujourd’hui Netscape Navigator devenu Firefox a repris de manière significative des parts de marché à Internet Explorer.

3.1. Les mécanismes de création des logiciels libres

Le modèle de création du Logiciel Libre s'appuie sur un fort investissement humain et sur le partage de valeurs au sein d'un même projet. Les projets rassemblent des utilisateurs et des développeurs aux savoirs-faire complémentaires organisés en communautés interagissant autour de systèmes de production de code, de documentation et de mises à jour. Ces organisations regroupent des utilisateurs du monde entier et des équipes pluridisciplinaires, comprenant aussi bien des industriels à travers la publication de recherches et d'outils expérimentaux que des éditeurs ou sociétés de services qui consacrent une part significative de leur chiffre d'affaires en R&D, des administrations ou encore des contributeurs individuels.

[10] Cet épisode est mentionné dans « The Cathedral & the Bazaar » de Eric Raymond, déjà cité.

Ce modèle de création communautaire favorise la démocratisation et la diffusion de l'innovation en constituant un patrimoine universel de connaissances, de méthodes et d'outils logiciels accessibles à tous.

L'efficacité de ce modèle réside dans une rupture par rapport aux modèles antérieurs. L'accélération des flux de développement repose sur une standardisation partagée entre les participants de ces communautés au niveau administratif, juridique et technique. Ces facteurs combinés sont des catalyseurs de la création de projets mondialisés. La réduction des frottements dûs aux rigidités de l'informatique traditionnelle permet de mutualiser la R&D à l'échelle de l'Internet, de faire évoluer les projets de manière continue, quotidiennement, dans un contexte international. La liberté de participer, de contribuer et d'utiliser localement les retombées d'un projet sans contrainte artificielle crée un catalyseur de valeur.

Il en résulte une accélération de l'innovation qui fait le succès professionnel et grand public des logiciels libres.

Le Logiciel Libre est issu d'un « laboratoire planétaire » volontaire et spontané.

3.2. Les communautés

Les acteurs contribuant aux logiciels libres sont des professionnels de l'informatique, des utilisateurs, des chercheurs, etc. Ils peuvent intervenir dans les projets au titre de leur activité professionnelle ou en dehors, par exemple en signalant bénévolement des défauts logiciels.

Ces acteurs réunis au moyen d'outils de communication issus d'Internet forment des communautés, c'est-à-dire un maillage relationnel et technologique qui permet à chacun de participer à l'évolution d'un projet en conservant son identité propre. Ces communautés forment des structures agiles permettant à tout un chacun de contribuer et de bénéficier des retombées du projet.

Les projets s'organisent en communautés

Des projets d'envergure locale ou nationale se constituent autour d'outils de collaboration simples de type forge logicielle ou listes de diffusion...

Au-delà d'une certaine taille et lorsque le cadre devient international, les grands projets utilisent leur propre infrastructure et s'organisent selon des statuts permettant la collaboration effective d'acteurs majeurs de l'informatique aux côtés de primo-utilisateurs : des associations ou des fondations, par exemple. C'est le cas pour des grands projets connus tels Linux, GNU, OpenSSH, Apache, Debian, PHP, Mozilla, Perl, OpenOffice.org, Eclipse, Java , Python, Ruby, devenus des standards de l'informatique.

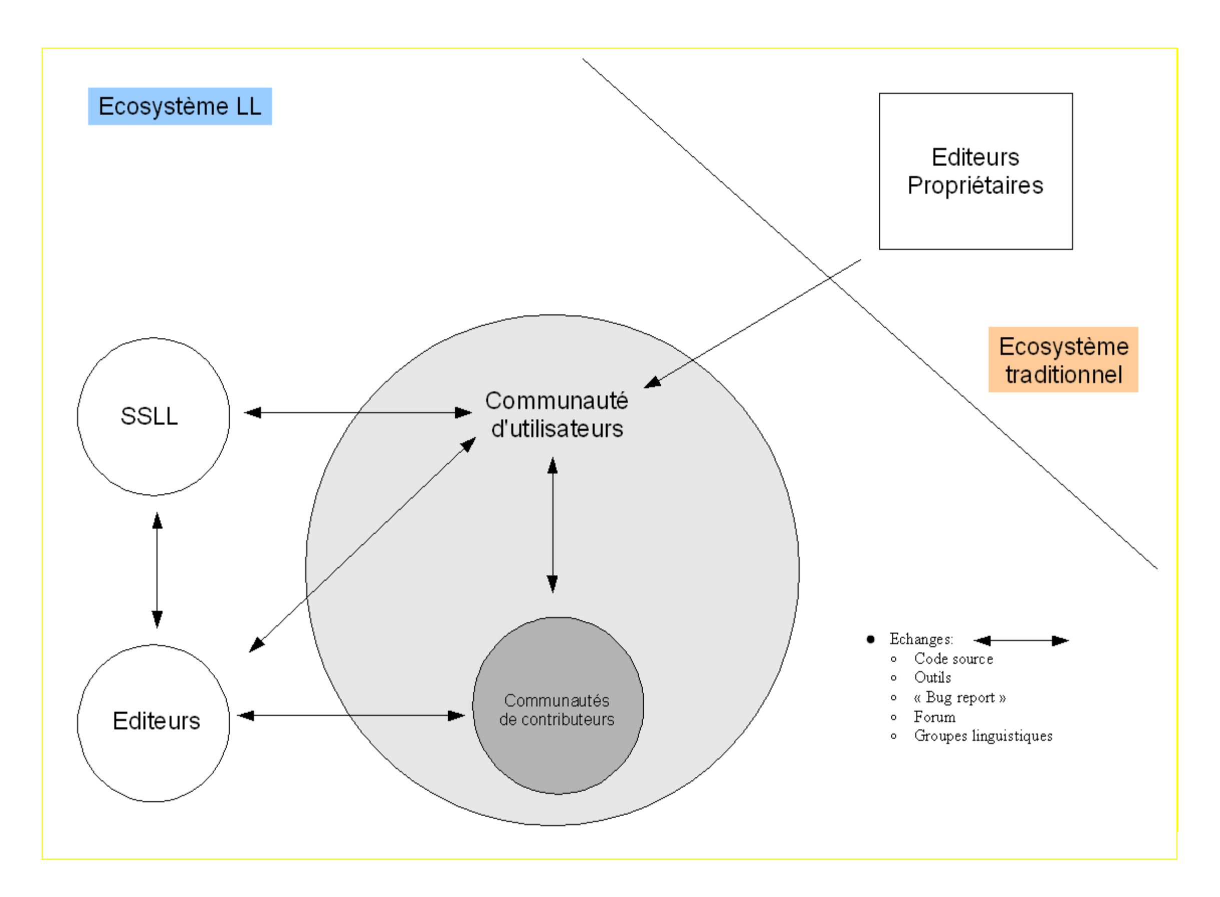
La protection juridique est garantie par des licences logicielles qui forment un socle assurant une bonne stabilité des réalisations, la simplification des processus juridiques et contractuels ainsi qu'une protection des œuvres créées. Le Logiciel Libre apporte un cadre juridique accepté par tous et assure une pérennité de l'investissement commun (*cf. «Quel cadre juridique pour les licences de logiciels libres ?»*).

Il est à constater que les logiciels libres n'ont généré qu'un très petit nombre de conflits juridiques malgré leur large diffusion.

L'efficacité des communautés : un avantage économique et juridique

Le dynamisme des communautés se traduit en avantages concurrentiels et économiques pour les entreprises dont l'activité repose sur le Logiciel Libre. Notamment, l'usage du Logiciel Libre permet de réduire « le time to market » (« temps de mise sur le marché ») d'une solution, et d'améliorer la productivité de l'économie numérique en s'insérant dans un marché mondialisé à forte réactivité pour un coût organisationnel minimal.

Les acteurs du secteur s'appuient sur des modèles économiques différents par type d'activité :



L'écosystème générique du Logiciel Libre

Une grande diversité d'acteurs professionnels

Le secteur professionnel dont l'activité est basée sur le Logiciel Libre ne cesse de s'agrandir : éditeurs, sociétés de services généralistes ou spécialisées, fournisseurs de services en ligne, commerçants en ligne, grands comptes, laboratoires de recherche, instituts de formation, industriels et constructeurs informatiques. Le Logiciel Libre permet également aux acteurs qui n'y contribuent pas directement de participer à son essor, en diffusant largement son usage. On peut par exemple noter le succès croissant des applications Firefox et OpenOffice.org aussi bien en entreprise que chez les particuliers ou dans les services publics.

La diversité des acteurs professionnels qui interviennent dans le monde du Logiciel Libre favorise son adéquation aux besoins du monde de l'entreprise.



3.3. Un continuum allant des métiers d'éditeur au service

L'expérience cumulée des adhérents de Telecom Valley et de l'April permet de répartir les modèles économiques du Logiciel Libre sur un plan selon 2 axes principaux :

- Un axe allant du service à la recherche et développement, en passant par le développement d'applications et de composants réutilisables, l'adaptation, l'installation, le support et la formation.
- Un axe représentant la spécialisation et la valorisation des applications logicielles en jeu, du niveau commodité jusqu'aux applications métier.

3.4. Vers une nouvelle chaîne de valeurs...

La création de valeur dans le Logiciel Libre est liée à la rapidité de mise à disposition des solutions, de leurs évolutions, ainsi qu'à la prise en compte des possibilités offertes par les nouvelles méthodes de travail collaboratif. Les acteurs professionnels du marché se situent sur un continuum allant du métier d'éditeur au service

Des entreprises industrielles dont l'activité principale s'écarte de ces modèles (constructeurs, fournisseurs de services en ligne, utilisateurs) emploient du Logiciel Libre pour des usages banalisés à un coût faible (commodité) et pour des situations spécialisées ou complexes (applications métier).

Tous ces acteurs s'organisent en une chaîne de valeur. Ils constituent un écosystème diversifié pour produire une activité économique en plein essor : celle du Logiciel Libre dont le moteur principal est la dimension communautaire autour de la valeur ajoutée.



4. LE LOGICIEL LIBRE, VECTEUR DE CRÉATION DE VALEUR

Cette partie vise d'une part à définir l'ensemble des apports des logiciels libres du point de vue technologique, économique, de l'innovation, de leur diffusion, de leur appropriation et de leur notoriété d'autre part, à présenter la valeur ajoutée pour le consommateur de l'informatique, l'utilisateur final, les industriels de l'informatique et pour l'ensemble de l'industrie.

4.1. Les apports généraux des logiciels libres

Les logiciels libres sont accessibles à tous. Tout le monde peut y contribuer, les utiliser en tout ou partie, ou créer un marché. Cette liberté conduit à des éléments différenciateurs par rapport au modèle traditionnel :

- Le ticket d'entrée est faible, les licences sont souples, ce qui rend le marché plus accessible pour de nouveaux entrants, sans limite à la volonté ou l'imagination de chacun ;
- L'adaptabilité des logiciels libres leur confère une capacité de répondre aux besoins des utilisateurs en toute indépendance technologique, aux besoins des développeurs sans les limites imposées par un calendrier unique d'éditeur ;
- Plus de transparence dans le choix de son partenaire et de sa plate-forme. Ce modèle oblige en outre les partenaires choisis à rester compétitifs et performants puisqu'ils sont jugés en fonction de la valeur ajoutée qu'ils produisent (adhérence stricte aux standards de l'industrie - protocoles, formats de documents).
- Une saine compétition entre les technologies, les projets ou les acteurs du Logiciel Libre tirent vers le haut le nombre et la qualité des solutions offertes.

Les logiciels libres permettent à l'industrie informatique de sortir de la relation classique « client/fournisseur » pour ouvrir d'autres possibilités de collaboration entre acteurs, en particulier autour du principe de communauté de développement, au service de la productivité et de l'innovation. Cette transparence contribue à élever le niveau général en réétalonnant le marché vers le haut.

4.2. Les extensions de la technologie

La publication dans les communautés-projets sur l'Internet permet un accès généralisé à la technologie, une meilleure diffusion du savoir. Globalement, l'adoption des technologies subit un double phénomène d'accélération : la baisse du coût d'accès au savoir et le levier de la distribution à travers le réseau mondial. La valeur apportée aux utilisateurs bénéficie ainsi également d'une diffusion plus rapide des technologies et des dernières évolutions.

Prenons l'exemple de la contribution de Sun Microsystems à la communauté OpenOffice.org avec le développement d'une boîte à outils Open Document Format (ODF) visant à séparer les outils de bureautique de la gestion des documents. Avec le développement et la promotion de ce nouvel outil, l'éditeur prouve sa capacité à répondre aux besoins de ses clients tout en permettant à la communauté des logiciels libres de créer de nouvelles fonctionnalités ou utilisations et les opportunités offertes par ce cercle vertueux de collaborations croisées sont pour l'ensemble des parties prenantes (Sun Microsystems, communautés, utilisateurs) aussi nombreuses que les volontés respectives de chacun.

Les standards ouverts apportent souvent une contribution essentielle à l'adoption des logiciels libres. Réciproquement, les logiciels libres sont, dans bien des cas choisis, comme implémentation de référence de ces mêmes standards et la transparence d'accès (gratuité, code source fourni) favorise leur très large diffusion (exemple de la standardisation du protocole DNS avec Bind, ou du courriel avec Sendmail).

Le mode de développement Logiciel Libre conduit ainsi à financer le développement de composants logiciels ou la participation de journées d'experts qui sont remis en commun pour partager les résultats. Il n'y a plus d'investissement dans la refonte de commodités mais exclusivement dans les évolutions de la technologie : composants, bibliothèques, nouveaux modules fonctionnels...

La valeur ne se situe plus dans la culture du secret de la maîtrise d'un code, mais dans la capacité à communiquer, à construire et à fédérer une communauté, à recruter de nouveaux talents, puis à générer des besoins en services additionnels. Un utilisateur n'achète plus des logiciels de base. Il achète des modules spécifiques, de l'expertise métier ou de la mise en œuvre couvrant exactement ses propres besoins. Le modèle repose sur la création de valeur ajoutée et non plus sur une économie de rente.

L'ouverture du code source offre à la fois une garantie d'intégrité des solutions de développement autour de standards communs, un gage d'interopérabilité pour les systèmes complexes, ainsi qu'un gage de pérennité des solutions choisies. Le Logiciel Libre apporte en ce sens une réponse fiable aux industriels dans les secteurs sensibles où l'indépendance technologique est indispensable (défense, administration, énergie). On peut notamment citer l'architecture Copernic de l'administration fiscale française - qui fournit entre autres le service de déclaration en ligne pour les impôts sur le revenu-, bâtie sur des composants logiciels libres.

Dans ce marché mondialisé, la généralisation du Logiciel Libre ouvre de nouveaux marchés et fait émerger de nouveaux talents. L'échelle de valeur est basée sur l'émulation autour de technologies innovantes.

4.3. L'accélération de l'innovation

Le partage des efforts de recherche autour des standards ouverts contribue à en améliorer l'interopérabilité et l'adoption à l'échelle mondiale. Les logiciels libres fournissent aujourd'hui les briques essentielles à la fabrication des systèmes et des applications. Une fois libérées du poids de la réécriture des fonctionnalités de base de leur système d'information, les entreprises peuvent ainsi se concentrer sur leur métier et libérer des ressources soit pour la recherche, soit dans de nouveaux développements, soit pour l'amélioration des performances. D'une manière générale, la compétition qu'installe le code ouvert pousse les acteurs du marché à se différencier par une innovation constante sur des cycles courts.

Par exemple, la libération des sources et spécifications du processeur SPARC a abouti à des versions de masque à 65 nanomètres par l'université de Santa Cruz (États-Unis). La publication de ces travaux permet à l'ensemble de la communauté (et donc à l'industriel Sun Microsystems) de partager un savoir-faire unique au monde et à en améliorer les performances.

Au niveau des commodités, l'excellence technique et l'innovation technologique dans le Logiciel Libre ont conduit à concevoir les systèmes les plus fiables (GNU/Linux, BSD, OpenSSH) vers lesquels converge le monde Unix, ainsi que les principales couches middleware (Python, Java, JBoss, ObjectWeb). Les projets complexes sont aujourd'hui basés sur du Libre : les produits à fort besoin de rendement et de fiabilité (virtualisation, terminaux ADSL, portails et moteurs Internet, VOIP, etc.), la téléphonie et l'embarqué massivement développés sur des noyaux libres, ou le monde de l'ASP.

Les logiciels libres sont énormément utilisés dans des domaines de recherche de pointe comme le calcul haute performance (HPC, le CERN déploie des grilles de calcul basées sur des architectures en Logiciel Libre), l'industrie aérospatiale (la NASA utilise Java pour piloter des sondes spatiales sur Mars), la recherche génétique, ou la couche sémantique du web. Ils bénéficient ainsi des dernières avancées sur le plan algorithmique ou mathématique, gage de performance et de fiabilité.

Les méthodes de collaboration du Logiciel Libre sont elles-mêmes retenues dans les laboratoires du monde entier comme modèle de partage : forums, wikis, P2P, forges logicielles, gouvernance des communautés, etc.

De nombreuses actions de recherche fondamentale dans des domaines comme la parallélisation des codes, la gestion de la mémoire des machines virtuelles, la prise en compte du « temps réel » applicatif ou les méthodes de refactoring de code de grande dimension sont réalisées en mode « Libre ». Ces travaux initiés par les industriels en collaboration avec le monde universitaire et les communautés sont au cœur d'une synergie aussi unique que prometteuse dans l'histoire de l'informatique.

5. DES MODÈLES ÉCONOMIQUES

Il s'agit dans cette partie de présenter le continuum allant du métier d'éditeur au service à la base de la création de nouveaux modèles économiques : les modèles économiques du Logiciel Libre.

5.1. Marché

L'émergence du marché du Libre et de ses acteurs

1991 voit émerger le noyau Linux et son modèle collaboratif. Son évolution a conduit à valider des technologies désormais utilisées dans le monde entier. Étroitement associées à l'explosion des réseaux, DNS Bind, Sendmail, Apache, Linux, BSD, GNU, Perl, PHP sont devenus les standards ouverts de l'Internet.

Ce déploiement a été émaillé de success stories telles que celles de Cobalt, Star Division, SleepyCat (pour BerkeleyDB), ADACore (pour le compilateur ADA GNAT) ou SuSE Linux. Plus récemment, JBoss racheté par Red Hat 331 millions de dollars, SourceFire introduite avec succès sur le Nasdaq et Red Hat valorisée 6 milliards de dollars sur ce même marché, soit plus de 12 fois son chiffre d'affaires annuel.

Les premiers acteurs de la diffusion du Libre ont été les éditeurs de distribution GNU/Linux : Red Hat sur le marché américain (1995), SuSE en Allemagne (1996), Mandrakesoft en France (1998 – maintenant devenue Mandriva), et Debian en tête des distributeurs communautaires à partir des années 1990.

Ces sociétés proposent de distribuer le système GNU/Linux associé à un ensemble de logiciels libres pour installer un serveur ou un poste de travail. Leur modèle économique est centré sur la vente d'offres additionnelles, de formation et de support. En parallèle, entre 1996 et 1999, ils ont favorisé l'essor des sociétés de services spécialisées qui sont apparues pour proposer des prestations de mise en œuvre de logiciels libres : développement, intégration et support.

Les pionniers du service en France sont Alcôve (1996) et Easter-Eggs (1997). Elles ont créé le modèle de la société de services spécialisée en logiciels libres. Ce concept sera plus tard repris par d'autres entreprises telles que Aurora (1999), Idealx (1999), ou encore Open Wide, Nuxeo et Linagora. Désormais, ce modèle est repris par les entités spécialisées créées dans les principales sociétés de services généralistes du marché.

GNU/Linux, valeur émergente du marché informatique

Fin 2000, IBM annonce 1 milliard de dollars d'investissements « autour de Linux » et sonne le ralliement des grands comptes. L'ensemble des éditeurs et constructeurs informatiques rejoignent le mouvement et investiront au fur et à mesure sur la plate-forme GNU/Linux : elle émerge en tant que valeur du marché de l'informatique.

Depuis, le modèle Libre a largement prospéré et permis d'accélérer la diffusion d'innovations notables dans les moteurs de recherche (Google, Voila), la sécurité informatique, la programmation objet (Java, Python, Perl, PHP), la téléphonie (Java), etc.

De grandes administrations sont aujourd'hui clientes du Logiciel Libre aux États-Unis, en Europe, dans les pays émergents, et pour la France, les principaux ministères (Intérieur, Minefi, Agriculture, Éducation nationale notamment), le secteur public (gendarmerie, CEA, CNRS) ainsi qu'un nombre

important de grandes entreprises. Récemment, le Groupe PSA a annoncé basculer 20 000 postes de travail vers des postes utilisant le système GNU/Linux .

5.2. Modèles économiques

Définition du continuum

L'émergence du Logiciel Libre est liée à l'accélération du marché des technologies de l'information et au besoin de compétitivité des entreprises.

Pour rappel, les facteurs clés sont :

- le développement fort de l'Internet qui joue le rôle de catalyseur,
- l'externalisation croissante des systèmes informatiques,
- le besoin de productivité,
- le développement des logiciels libres.

Le modèle Libre a permis l'émergence dans le monde professionnel de nouvelles entreprises aux modèles hybrides. Ces modèles modifient en profondeur les relations entre éditeurs, distributeurs et sociétés de services.

Les « éditeurs » tirent leurs revenus des besoins de standardisation recherchés par les clients, tant dans les logiciels que dans les besoins d'évolutions.

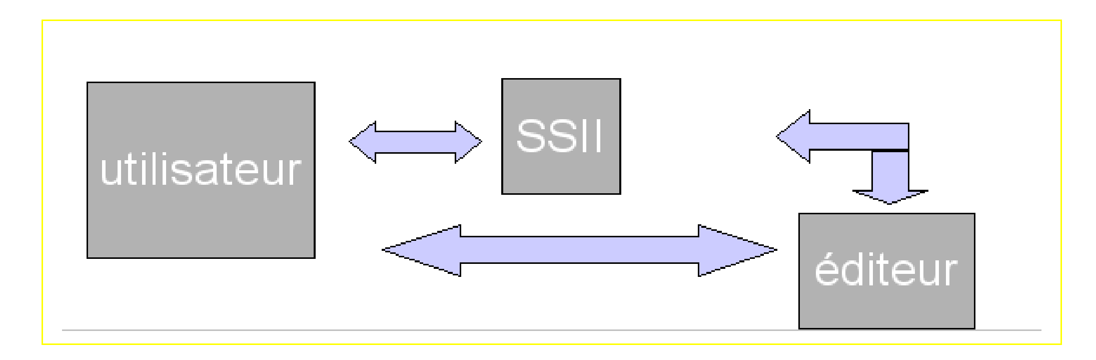
Les distributeurs fournissent des suites complètes de logiciels en boîte blister ou en « appliance ». Ils vivent de la formation et du support.

Les « SSII » développent des logiques d'offres associant logiciels et services orientés métiers liées à leur savoir-faire. Elles vivent du déploiement et de l'intégration.

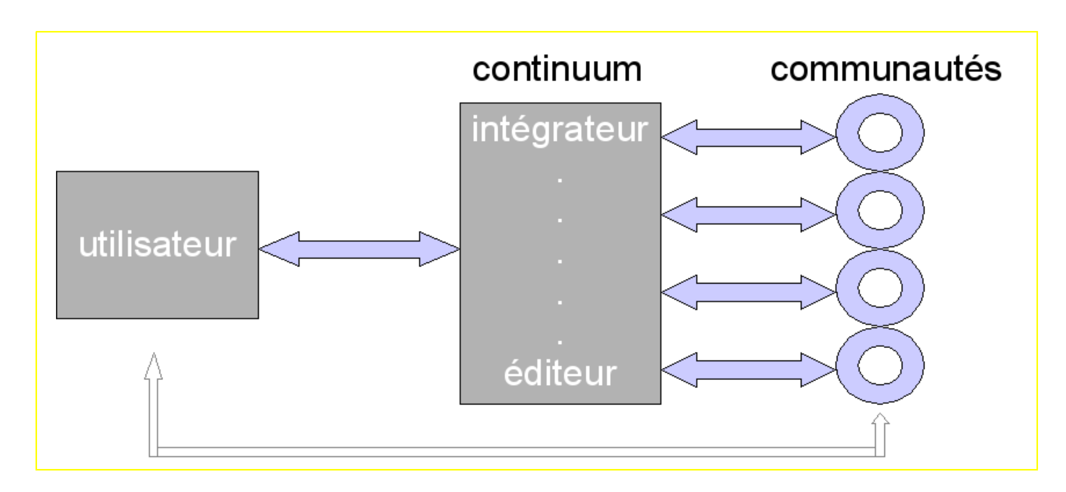
Le Logiciel Libre a redonné à l'ingénierie une nouvelle vigueur en mettant à disposition tout un ensemble de technologies fortement paramétrables et de composants réutilisables, personnalisables, qui sont autant de commodités en libre accès sur Internet. Ceci amène une montée inéluctable de la valeur ajoutée associée au service. Une tendance soutenue par l'amélioration des outils (gestion de projets, pilotage...). À côté de cela, un rééquilibrage du contrôle de la technologie se produit en faveur des clients finaux qui réclament davantage de « progicialisation » ainsi que des modes de consommation à la demande (SOA, SAAS) pour industrialiser les déploiements et la maintenance. Cela conduit à concilier innovation, standardisation, distribution, personnalisation (spécifique), et retour sur investissement. D'où la spécialisation progressive des sociétés du secteur en faveur de modèles hybrides d'édition de logiciels, de distribution et d'intégration spécifique très orientés sur une économie de services. Face à cela, les acteurs de l'informatique propriétaire (éditeurs, intégrateurs...) doivent ouvrir leurs offres et adapter leurs modèles économiques en recréant de la valeur sur l'innovation et non plus sur la « rente ».

La professionnalisation du marché du Libre s'inscrit dans une logique de continuum dans laquelle les modèles « éditeurs » et « services » se rejoignent pour tendre vers des modèles hybrides adaptés aux modes de consommation des clients et des enjeux marché du moment.

La chaîne relationnelle de l'industrie du logiciel propriétaire :



La chaîne relationnelle de l'industrie du Logiciel Libre :



Y a-t-il des modèles spécifiques au Libre ?

On peut considérer qu'il existe des modèles spécifiques au Logiciel Libre en ce sens qu'ils intègrent la réalité liée au changement de relation entre les éditeurs et les «SSII ».

Au départ éditeurs de distribution (Red Hat, Mandriva, SuSE Novell) et sociétés de services spécialisées dans la mise en œuvre de logiciels libres (les SSLL), modèles de type hybride, constituaient la chaîne de valeur de l'industrie du Libre. Désormais, plusieurs facteurs viennent modifier cette chaîne de valeur :

- la taille des projets rend les besoins en services plus importants,
- l'étendue des projets dans le système d'information étend les besoins en compétences diversifiées,
- le choix de logiciels libres dans des projets critiques induit de forts besoins en maintenance et en garantie,
- les commodités ayant fortement évolué, les besoins en progiciels métiers et offres à fort contenu technique entraînent des besoins d'évolution et d'innovation technologique constants.

Ainsi, les professionnels du monde Libre participent en tout ou partie à cette nouvelle chaîne de valeur du modèle Libre, autour de plusieurs métiers :

- La conception et le développement de composants et d'outils libres = métier d'éditeurs technologiques

- et/ou, la distribution et la commercialisation de ces composants/outils = distributeurs et éditeurs commerciaux
- et/ou, les prestations des SSII = conseil, AMOA, AMOE, Ingénierie (régie, forfait), Formation, Maintenance (TMA)
- et/ou, les services additionnels éditeurs = Support, Mises à jour (certification, documentation, évolutions, formation…)
- et/ou, les apports de l'écosystème du Libre = contribution aux communautés, diffusion des innovations.

Leur modèle reste fortement orienté sur différents types de prestations de services, cependant leur activité est davantage orientée côté utilisateur ou côté technique.

Le partage de valeurs et de pratiques communes permet de définir les entreprises du Libre :

- la veille technologique couplée avec une interaction avec les communautés du Libre,
- le respect du modèle de développement Open Source et des standards de professionnalisme et d'éthique du Logiciel Libre,
- la compréhension, l'utilisation et la diffusion des licences libres.

Typologie des modèles économiques du Libre

En fonction de leur positionnement par rapport à la chaîne de création de valeur du Libre, les professionnels se différencient entre eux par les modes de commercialisation de leurs offres auprès de leurs clients. Nous avons ainsi recensé plusieurs modèles économiques qui reprennent dans leur modèle de vente tout ou partie des spécificités métiers de la chaîne de valeur du Libre :

- le modèle d'éditeur (licence double ou licence décalée),
- le modèle distributeur à valeur ajoutée,
- le modèle ASP,
- le modèle de services à valeur ajoutée,
- le modèle d'intégrateur hybride.

Modèle d'éditeur de double licence ou licence décalée

Ce modèle concerne tout autant des éditeurs qui révolutionnent leurs modèles économiques (Ingres, Sun) que de nouveaux éditeurs 100% Libre (Mysql, AdaCore, Wallix) qui développent de nouveaux composants et/ou outils (frameworks). Ils proposent de publier dans la Communauté une version standardisée et mature de leur offre, afin de prendre position sur leur marché. Ils continuent d'investir sur de nouvelles versions vendues sous forme d'offres additionnelles ou de maintenance autour de leur noyau de base.

En parallèle, certains proposent une offre supérieure en fonctionnalités aux clients qui le demandent sur la base d'une tarification traditionnelle, et d'un autre modèle de licence. Imposer un standard par l'ouverture du code constitue l'originalité de ce modèle créateur de valeur dans la mesure où l'accessibilité est offerte au client ainsi que le prototypage potentiel. Nous sommes bien là dans un nouveau modèle issu de la révolution Logiciel Libre, à travers l'ouverture du code source et le fait d'appliquer des idées innovantes non pas pour protéger un fond de commerce, mais pour développer de nouveaux marchés. Une variante (le modèle à licence décalée) consiste pour un éditeur, sur un principe équivalent, à ne facturer que les versions récentes de ses logiciels (6 à 12 mois) et ensuite à les publier dès commercialisation d'une nouvelle version majeure. Le client ne paye que pour l'innovation et éventuellement le support. Les éditeurs vivent des services produits (maintenance logicielle, support revendeur, assistance utilisateur), et de la vente de modules additionnels.

Modèle de distribution à valeur ajoutée (Mandriva, Red Hat)

Ce modèle recense les acteurs dont le métier consiste à assurer la distribution physique de logiciels libres au sein d'un système d'exploitation complet rassemblant des versions sélectionnées disponibles sur le marché, et autour d'outils de mise en œuvre et d'utilisation adaptés aux besoins de leurs clients. Ils proposent une partie des services additionnels attendus par les clients autour du support et de la formation aux logiciels.

Lors de l'achat de logiciels propriétaires, l'utilisateur acquitte un droit d'utilisation personnel sous forme d'une licence payante ainsi qu'une redevance correspondant à la maintenance pour une durée d'un an. Lors de l'acquisition de logiciels libres auprès d'un distributeur, l'utilisateur a le choix entre télécharger son produit ou acheter une boîte classique incluant un cédérom et une documentation. Un abonnement à la maintenance permet d'accéder au système automatique de mises à jour, ainsi qu'aux services variés tels que l'assistance technique ou la formation nécessaires. L'utilisateur est libre de consommer selon ses besoins.

Les distributeurs génèrent des revenus de la vente des supports physiques de leur système d'exploitation en grande quantité, de la vente de services à valeur ajoutée : formation et support technique, ainsi que de la revente de leurs produits à travers des réseaux de revendeurs ou des accords OEM (Original Equipment Manufacturer).

Modèle ASP

Le Fournisseur de service d'application (Application Service Provider - ASP) est une entreprise qui fournit des services informatiques à ses clients au travers du réseau Internet. Le point fort de ces entreprises est de fournir un accès à des applications particulières (comme un programme de facturation médicale) en utilisant un protocole standard comme HTTP. En français, le terme FAH est aussi utilisé, il signifie Fournisseur d'Applications Hébergées. Ce modèle répond à une forte demande du marché de disposer en ligne d'applications complètes et opérationnelles répondant à un besoin spécifique précis sous forme de location ou d'abonnement. Sa valeur ajoutée réside essentiellement dans le « on ne dépense que ce que l'on consomme », et dans son accessibilité.

Les ASP tirent avantageusement parti du Libre pour réduire leurs coûts d'infrastructure, tout en maîtrisant la distribution et la technologie qu'ils proposent à leurs clients.

Les spécificités des ASP sont prises en compte à travers des licences spécifiques comme la GNU Affero GPL.

Modèle de service à valeur ajoutée

Dans ce modèle, nous retrouvons les sociétés qui offrent l'ensemble des prestations de services d'une SSII (conseil, ingénierie, formation, TMA...) en s'appuyant sur une palette d'outils et de composants spécifiques du Libre. Souvent, ces entreprises ont une spécialisation fonctionnelle (CMS, CRM, mobilité) ou métier (ERP, informatique industrielle, sécurité).

Leur valeur ajoutée repose à la fois sur le savoir-faire de leurs experts impliqués dans les communautés du Libre, sur leur forte spécialisation technologique et sur leur connaissance fonctionnelle métier.

Ce modèle repose sur la vente de prestations intellectuelles sous toutes ses formes : conseil, expertise, développement au forfait, en régie, TMA. Ce modèle est recensé sous le concept marketing de SSLL.

Modèle d'intégrateur hybride

Ce modèle doit favoriser la nécessaire synergie inhérente au Logiciel Libre entre les métiers d'éditeur et ceux du service pour le seul profit du client, notamment dans les grands comptes et administrations.

On entend par « hybride » le fait de fédérer des offres « produits » et des offres « services » au sein d'un prestataire unique positionné soit sur un axe métier (par exemple la sécurité), soit sur l'axe du système d'information global de l'entreprise (par exemple le système d'information d'une collectivité locale, d'une filiale ou division d'une entreprise du CAC 40) en garantissant au client la cohérence et la pérennité de l'ensemble.

Ce modèle implique la recherche d'une taille critique minimale adaptée aux marchés des grands projets, sans « encapsulation » dans les offres des intégrateurs et opérateurs traditionnels.

Par là même, ce modèle facilite le choix pour un DSI de mettre en œuvre des alternatives en Logiciel Libre reconnues et matures sur le marché.

Les revenus de l'intégrateur proviennent de la revente des technologies et des prestations de mise en œuvre chez les clients.

Comment va évoluer le marché d'ici à 2012 ?

Jusqu'à présent, les principales caractéristiques du marché du Logiciel Libre étaient celles d'un marché émergent dans le secteur des technologies de l'information, à savoir :

- une forte croissance de + 30% en moyenne jusqu'à cette année (source Pierre Audouin Consultants), en passant de 0,5% à 5% du marché global des technologies de l'information,
- une multiplicité d'acteurs de petite taille,
- des acteurs avec des stratégies de niches et des modèles économiques de type « tout en un » adaptés à des tailles de petits projets (SSLL),
- des investisseurs financiers dans des logiques de « capital risque »,
- le rôle important et nécessaire sur le plan médiatique des pionniers faiseurs du marché, et un positionnement marketing comme seul facteur différenciateur,
- une communication très idéologique, et donc fortement liée aux pouvoirs politiques et/ou aux sponsors industriels qui investissent dans le secteur soit pour des motivations d'indépendance, de sécurité, soit par mesure d'acquisition ou de protection de parts de marché,
- le recours à la sous-traitance comme modèle économique ou aux stratégies de subventions directes ou indirectes auprès des pouvoirs publics,
- peu de croissance de chiffre d'affaires, forte notoriété.

D'un marché confidentiel et émergent, le Libre est en train de pénétrer toutes les couches du système d'information et entre dans une phase de maturité, de structuration et de consolidation.

L'arrivée brutale de la crise à l'automne 2008 a pour corollaire un regain d'intérêt des entreprises pour la mise en oeuvre de solutions Open Source. Plusieurs études de marché publiées début 2009 confirment cette tendance.

D'un modèle « tout en un » (SSLL), capable de proposer l'intégralité de la chaîne de valeur sur des projets de faible envergure, ou de distributeur de commodités systèmes de type GNU/Linux accessibles en téléchargement, les acteurs du Libre doivent s'adapter à une demande plus globale, à des tailles de projets plus importantes ainsi qu'à la professionnalisation du marché.

Cette phase de structuration et de consolidation passera par :

- un remodelage des métiers et des modèles économiques des acteurs du Libre, capable de générer des modèles intermédiaires en phase avec les différents cycles de vie des acteurs concernés. Par exemple, un modèle de type distributeur à valeur ajoutée peut temporairement s'inscrire dans un schéma de type service à valeur ajoutée pour assurer son développement autour d'une première offre, en attendant la mise à disposition de nouvelles offres technologiques...
- une phase de consolidation qui peut être opérée :
    - sur un axe vertical/métier (sécurité, commodités...) et donc, potentiellement, par un acteur de type distributeur à valeur ajoutée ou un éditeur de logiciel, interne ou externe au Libre,
    - sur un axe de type intégrateur hybride, spécialisé dans le Libre et à même de traiter des projets de grande taille et à plus forte criticité pour les clients,
    - sur un modèle industriel plus classique sous forme d'un prolongement d'activité dans une offre de SSII qui se positionne sur le marché avec une offre complète où cohabitent technologies libres et non libres.

Et la nécessité pour ce marché du Libre de se doter de champions industriels 100% libre, d'une taille critique et d'un niveau de rentabilité qui pérennisent le modèle économique du Libre. Acteurs qui seront concurrentiels par rapport aux acteurs informatiques traditionnels (constructeurs, éditeurs, et intégrateurs...).

6. LES ENJEUX TECHNOLOGIQUES, ÉCONOMIQUES ET STRATÉGIQUES

Deux facteurs ont facilité l'essor du Logiciel Libre : la baisse du coût des ordinateurs et le déploiement d'Internet.

D'autres phénomènes actuels accélèrent l'usage de Logiciel Libre : l'évolution des technologies entraîne un accroissement des équipements grand public contenant une informatique embarquée de plus en plus puissante et communicante.

Les enjeux de l'économie numérique dans la vie courante favorisent ainsi une capillarité de plus en plus importante des logiciels sous forme de commodités.

6.1. L'enjeu du Logiciel Libre dans l'économie de l'immatériel

L'arrivée massive du numérique dans tous les secteurs économiques met en évidence les propriétés particulières des « biens immatériels ». Leur importance croissante et les évolutions qu'ils entraînent ont fait apparaître de nouveaux enjeux économiques autour d'une économie de l'immatériel. Les objets informationnels ont une propriété clé : le coût marginal d'une copie est presque nul. Jusqu'à l'apparition du numérique, l'économie a fonctionné principalement sur l'échange de biens matériels, dont la possession est exclusive. Cela suppose que lorsqu'un bien matériel est vendu, sa propriété passe à l'acquéreur.

Pour un bien immatériel, lorsque le bien est vendu, le vendeur peut encore le posséder (on parle de non-rivalité). L'acte de vente constitue en général une duplication du bien, ce qui ne demande pas d'opération de production lourde.

Le paradigme de la rareté des biens facilite le fonctionnement de l'économie traditionnelle. L'arrivée de l'Internet au contraire et de la diffusion de masse bouscule les modes de commercialisation préexistants tout en générant de nouvelles possibilités de développement économique. Les récents débats autour du téléchargement des œuvres musicales sur Internet et le vote houleux des lois DADVSI et HADOPI en sont bien sûr une autre illustration. La question du modèle de rémunération de la production des biens informationnels, notamment des logiciels informatiques, est dans ce contexte un enjeu majeur. Faut-il s'opposer aux évolutions offertes par Internet en finançant des barrières à la diffusion ?

Le Logiciel Libre suit la voie de la diffusion sans limitation, en proposant ainsi de nouveaux modèles économiques pour l'économie de l'immatériel.

6.2. L'enjeu économique de l'innovation

Face au désavantage de ne pouvoir baser son modèle économique sur un revenu tiré de la vente de chaque copie, le Logiciel Libre bénéficie d'atouts qui contribuent à son succès.

En premier lieu, le système des licences libres permet une adaptation rapide des projets à la mondialisation de l'économie et à l'accroissement des échanges numériques au niveau mondial, en particulier via Internet. La rapidité de réaction face aux besoins des clients et à l'évolution de la technologie constitue un avantage économique certain.

Le Logiciel Libre permet de monter des projets réunissant des structures de natures différentes, internationalement. Il permet aux participants d'entrer dans le projet ou d'en sortir à tout moment sans frais et avec une complexité contractuelle et juridique minime. Face à la difficulté de contractualiser une collaboration dans un contexte de logiciel propriétaire, il s'agit là d'un avantage en terme de réactivité qui peut se compter en mois, voire en années.

Ensuite, le Logiciel Libre facilite la mise à disposition de versions précoces du logiciel et la prise en compte du retour d'expérience des utilisateurs très tôt dans le cycle de développement. Ces deux caractéristiques permettent au Logiciel Libre de bénéficier en général d'un time to market inégalable par un schéma propriétaire. Cet avantage est d'autant plus net qu'il s'applique à des solutions innovantes, pour lesquelles avoir la première réalisation disponible sur le marché est primordial.

En outre, les modes de travail collaboratifs du secteur Logiciel Libre sont particulièrement adaptés au travail distant via Internet. Par exemple, l'innovation d'une fonctionnalité simple comme la génération d'un document PDF directement depuis le traitement de texte OpenOffice.org représente un avantage ergonomique que le leader du marché peine à rattraper en raison de la complexité contractuelle de cette fonctionnalité en environnement propriétaire.

L'innovation s'accompagne en parallèle d'une baisse relative de la valeur des solutions génériques (« commoditisation »). Que ce soit en tant qu'utilisateur ou fournisseur de solution, l'adaptation à cette dynamique du marché est facilitée par le déploiement de standards ouverts et de logiciels libres. Les solutions alternatives se multiplient et s'étendent maintenant au poste de travail.

6.3. L'accès à la compétition globale pour les PME innovantes

Pour de nombreuses sociétés de petite taille (particulièrement les éditeurs), les avantages du Logiciel Libre en terme de rapidité de mise en œuvre contrebalancent l'absence de rémunération à la copie. Le faible coût de diffusion du logiciel (téléchargement libre) permet à une petite structure innovante d'obtenir une reconnaissance internationale de son expertise. AdaCore et Mandriva sont des exemples de sociétés qui utilisent l'effet de coût marginal nul de la duplication et l'effet de levier de la diffusion via Internet de solutions logicielles. Un modèle économique couramment associé à ce type de développement est la vente de prestations de support dans le monde entier.

6.4. Mutualisation de la R&D des grandes organisations

Pour les grandes sociétés, la mutualisation par le Logiciel Libre de certains domaines de leur R&D représente un choix stratégique lorsque disposer de solutions exclusives n'est plus un enjeu concurrentiel fort. La demande de support technique, de suivi des versions (sécurité) et plus généralement d'expertise est présente quelle que soit la licence du Logiciel. Le Logiciel Libre incite l'éditeur qui fait ce choix à partager une partie de ce chiffre d'affaires avec des sociétés de service locales, plus proches du client, mais l'éditeur gagne en contrepartie un réseau de support élargi et peut se positionner sur l'expertise tout en bénéficiant d'une R&D mutualisée (notamment par la réutilisation de composants libres).

L'utilisation et le financement du Logiciel Libre constituent un axe de leur stratégie à long terme. Il est difficile aujourd'hui d'imaginer certaines applications sans Logiciel Libre (les « box » de l'ADSL, les composants réseaux embarqués, les architectures orientées service sur Internet). À ce titre, ces sociétés font partie de l'écosystème du Logiciel Libre. Les modèles économiques vont donc au-delà des modèles directs de type éditeur/service pour aller vers des modèles indirects de mutualisation de

R&D et de « coopétition », comme celui des FAI (Fournisseur d'Accès à Internet) concurrents mais utilisant les mêmes briques logicielles de base. Un code source produit dans le cadre de la R&D d'un organisme a le potentiel de promouvoir le financeur dans les communautés de développeur dans la durée. L'importance de l'« effet goodwill » généré et la visibilité accrue des financeurs au sein du monde de l'Internet participent au retour économique des investissements réalisés.

6.5. L'enjeu de la transparence et de l'interopérabilité

La numérisation croissante de leurs processus métiers et de leurs informations rend les grandes organisations de plus en plus dépendantes de leurs outils informatiques, et donc de leurs fournisseurs. Dans ce contexte, la demande de transparence sur les logiciels (accès au code, auditabilité, sécurité juridique...) augmente car les organisations veulent avoir des garanties sur le fonctionnement de leurs systèmes, et minimiser leur dépendance vis-à-vis d'acteurs externes.

Un point fort du Logiciel Libre est de fournir des applications dont la pérennité, les évolutions et les coûts ne dépendent pas d'un éditeur unique, ce qui peut devenir problématique lorsque le poids de l'éditeur est tel que ses clients ne peuvent plus influencer sa stratégie. Du point de vue économique, les coûts en jeu incitent les utilisateurs à comparer la valeur ajoutée des solutions, à se méfier des « boîtes noires » et à mettre leurs fournisseurs dans une situation de concurrence. Présenter une offre ayant des garanties de transparence et d'indépendance devient un avantage concurrentiel. C'est par exemple le choix de Sun Microsystems avec Java et Open Solaris.

La numérisation pose aussi le problème de la conservation et de la réutilisation des informations de l'entreprise sur le long terme. L'utilisation de formats ouverts, désormais clairement définis notamment dans la loi française (article 4 de la loi n 2004-575 pour la confiance dans l'économie numérique du 21 juin 2004 (J.O du 22 juin 2004, p. 11168), pour stocker l'information s'impose progressivement. En permettant une indépendance vis-à-vis de l'applicatif pour accéder aux informations, les formats standards améliorent la pérennité des données, et ouvrent la porte aux logiciels libres.

Le besoin d'interopérabilité lié au maillage croissant des échanges informatiques pousse à l'usage de solutions basées sur des normes et standards pour les interfaces et les formats de données, ce qui est encore à l'avantage du Logiciel Libre.

En synthèse, la croissance des échanges et du volume d'information géré par les entreprises entraîne une demande de transparence des outils et de mise en concurrence des fournisseurs. Administrations et collectivités sont également concernées au premier chef par ces enjeux. Cela se traduit par une demande d'ouverture des applications (code ouvert, méthodes standardisées) et une demande de standards interopérables (pérennité des données et facilité de communication). Cette demande bénéficie directement aux offres basées sur du Logiciel Libre.

6.6. Les enjeux d'indépendance

Le Logiciel Libre permet une forte maîtrise de son système d'information. Notamment, les décisions sur les évolutions des composants logiciels peuvent se prendre avec une plus grande liberté qu'en environnement propriétaire. On rompt avec l'effet domino des architectures propriétaires dans lesquelles le changement d'un maillon entraîne le plus souvent l'obsolescence de la chaîne complète (matériel, OS, application). Pour les structures dont le système d'information est un enjeu majeur en terme d'indépendance, le Logiciel Libre est un atout. Certains acteurs économiques utilisent le Logiciel Libre car l'indépendance qu'il permet vis-à-vis des fournisseurs permet d'obtenir un niveau élevé de secret industriel (Google maintient ses concurrents loin de ses infrastructures informatiques et salarie

de nombreux contributeurs à des projets majeurs). Les leaders des télécommunications, du matériel informatique et des éditeurs de systèmes d'exploitation contribuent activement sur les logiciels qui leur permettent d'échapper au risque de domination d'un éditeur unique. Tous ces acteurs participent directement ou indirectement aux projets libres majeurs via des fondations (Mozilla, Apache) ou en salariant des développeurs (OpenOffice.org, Python, Samba).

Les gouvernements de nombreux pays veulent accroître leur souveraineté sur le secteur informatique, et soutiennent le Logiciel Libre au travers de projets de recherches ou de prestations de service (France et Europe, Inde, Japon...). Les systèmes d'information sensibles des ministères peuvent être soumis à des contraintes d'indépendance vis-à-vis des fournisseurs (pérennité, continuité de service, neutralité de l'administration, indépendance de l'État et sécurité informatique).

Les enjeux français et européens

L'Europe a fait le constat de la place prépondérante des États-Unis dans la technologie (Internet, logiciels, matériels informatiques) et des investissements majeurs d'autres régions (l'Asie a investi massivement dans la recherche et les logiciels libres : 85 milliards de dollars sur 10 ans). Face aux enjeux du numérique en terme de croissance et d'indépendance, l'Union européenne a défini un axe stratégique fort sur l'économie de la connaissance (stratégie de Lisbonne).

Le Logiciel Libre et les standards ouverts (dont Open Document Format) permettraient dès maintenant à l'Union européenne de résoudre la grave question de l'interopérabilité des systèmes et des contenus parmi 27 pays et 23 langues officielles. L'Europe doit maintenant saisir l'opportunité d'un investissement dans la recherche en s'appuyant sur les logiciels libres pour diffuser et accélérer l'innovation, tout en restant autonome sur ses choix technologiques sans dépendance financière.

Le Logiciel Libre s'insère dans un renouveau industriel dans un contexte de développement durable, par la réutilisation de composants et la cohérence facilitée de politiques de développement économique aux niveaux européen, régional et local.

Dans ce contexte, la France doit maintenir les initiatives actuelles :

- collaboration des acteurs (associations, services, industriels et éditeurs),
- diffusion des apports du Logiciel Libre et création de thématiques propres au Logiciel Libre dans les pôles de compétitivité,
- usage du Logiciel Libre dans les contextes d'indépendance technologique et de sécurité nationale,
  usage du Logiciel Libre dans le cadre de la modernisation des systèmes de l'administration et des collectivités territoriales.

6.7. Les enjeux pour la recherche

Les logiciels libres constituent une très importante opportunité pour permettre aux acteurs du monde de la recherche de disposer de nouveaux moyens d'échanges et de communication, de faciliter l'accès aux meilleurs travaux pour les petites équipes en favorisant les synergies dans des domaines de pointe.

De nouveaux moyens d'échanges

La recherche est un domaine depuis longtemps mondialisé et pour lequel l'informatique est devenue un outil indispensable. Les logiciels libres permettent par exemple l'élaboration de projets mondiaux autour du HPC (calculs de haute performances) mutualisant les efforts de mise au point de leurs codes sources et le partage du temps de calcul (grilles).

Les logiciels libres permettent également aux industriels de communiquer plus efficacement sur les travaux de leurs laboratoires de recherche à destination de leurs clients et de spécifier avec eux et d'autres communautés des sujets communs, des outils ou des standards correspondant à des besoins réels d'utilisations futures. Ces échanges transparents sont aussi une garantie de pérennité des travaux de recherche financés par des fonds nationaux et de leur libre partage dans une optique d'augmentation de la valeur du « bien public ».

Enfin, l'ensemble des PME (dont un grand nombre de start-up) peuvent puiser dans cet immense réservoir de savoirs, de connaissances et d'outils, sans avoir à financer un investissement lourd, et disposer des ressources nécessaires aux innovations, synonymes de création de valeur économique et d'emplois.

Des synergies nombreuses

Le secteur du logiciel est en constante mutation et les logiciels libres doivent pouvoir s'adapter au mieux à ces changements. Il existe de nombreux domaines de recherche pour lesquels les synergies entre les industriels et les communautés du Logiciel Libre sont multiples et porteuses d'évolutions technologiques prometteuses.

Les modèles économiques des logiciels libres ont maintenant atteint une maturité certaine. Les logiciels libres sont de plus en plus utilisés aussi bien au niveau professionnel que personnel et un réel écosystème professionnel s'est constitué autour de ces nouveaux modes de distributions.

Les associations de professionnels ont d'ailleurs vu le jour récemment comme le CNLL[11] (Conseil National du Logiciel Libre).

Comme nous l'avons vu ci-dessus il n'existe pas un, mais plusieurs modèles économiques adaptés à un cas d'usage particulier. Gageons que la vitalité de cet écosystème permettra encore l'apparition de nouveaux modèles économiques dans les années à venir.

[11] http://www.cnll.fr

7. QUEL CADRE JURIDIQUE CONCERNANT LES LICENCES DE LOGICIELS LIBRES ?

C'est sans doute le principal enjeu. Ne pas montrer le code source de ses logiciels, c'est pour beaucoup le meilleur moyen de protéger sa propriété intellectuelle. C'est pourtant aussi le meilleur moyen pour déposséder quelqu'un de son propre travail. Sans pouvoir accéder au code, il n'est pas possible de vérifier qui a écrit telle ou telle partie, telle ou telle fonction. Bien entendu si chaque société était capable de conserver ses sources secrètement, comme Coca-Cola garde secret la formule de sa boisson fétiche, il n'y aurait pas de problème, mais vu les millions de lignes de codes détenues par chacune, vu le turn-over chez les programmeurs, comment une entreprise de logiciel peut-elle être certaine que sa propriété intellectuelle ne va pas être récupérée, réutilisée, pillée?

La réponse, qui va de pair avec le fait de ne pas donner accès à son code source, est de breveter les logiciels comme cela est possible aux Etats-Unis et au Japon.

L'utilisation d'un système de brevet est censée permettre d'assurer un équilibre entre les inventeurs et le reste de la société. En résumé nous pourrions dire que c'est “the publication of non-trivial ideas, with an explicit guarantee of continued availability, which seeks to protect the interest of the small players, for the overall betterment of society, as others learn from and improve on the original ideas”. Malheureusement, ce système qui marchait bien dans l'industrie n'a pas eu les effets attendus dans le monde du logiciel (Bessen & Hunt, 2004).

Les logiciels libres sont de plus en plus fréquemment utilisés par les entreprises. De nombreuses collectivités locales, des administrations, l'Assemblée Nationale leur font désormais confiance. Les licences libres, contrats régissant les rapports entre le concédant et le licencié, furent créées dans le but de s'opposer aux licences propriétaires que certains percevaient comme un processus de verrouillage de l'informatique. L'idée fondamentale de ces licences est de permettre à chaque personne qui le souhaite de travailler sur le logiciel de façon transparente : de l'utiliser, la modifier et la distribuer librement.

Ainsi, le code source, défini comme l'ensemble d'instructions écrites dans un langage de programmation informatique de haut niveau, c'est-à-dire humainement compréhensible, permettant d'obtenir un programme pour un ordinateur, est mis à la disposition du public.

Contrairement à une idée reçue, le recours à des logiciels « libres » ne signifie en rien qu'il n'y ait aucune obligation légale à respecter.

Cette multiplication des licences conduit à une réflexion nécessairement approfondie pour déterminer sous quelle licence il conviendrait de placer son œuvre. En effet, les problèmes de visibilité, de qualité juridique et de clarté du texte qui peuvent être très variables, constituent un véritable obstacle. C'est pourquoi l'utilisation de logiciels « libres » doit résulter d'un choix global prenant en compte, notamment :

- Les besoins en terme de fonctionnalités du logiciel ;
- Les coûts directs et indirects (intégration, maintenance et formation) ;
- L'interopérabilité avec les autres applications des systèmes d'information de l'entité ;
- La nature juridique du logiciel et la licence ;
- Les conditions d'utilisation du logiciel ;

- Les «suites» juridiques et économiques (développement, migration, intégration, maintenance, vente et distribution commerciale, etc.).

Ce choix doit également tenir compte du changement de modèle économique qu'entraîne l'utilisation généralisée de logiciels « libres ». La rémunération accordée à l'éditeur de logiciels propriétaires se trouve transférée vers les prestataires de services. Plusieurs grandes entreprises informatiques ont modifié leur stratégie en ce sens (ex : Sun, IBM). C'est pourquoi il est essentiel pour l'éditeur de logiciel, dans l'optique de se placer sous une licence de logiciel libre, d'assurer un suivi logistique, une sorte de service après vente permettant à la fois d'être rémunéré, mais qui présente également l'avantage de fidéliser l'utilisateur.

Le fait de disposer d'un logiciel « libre » présente **l'avantage pour l'utilisateur d'obtenir un logiciel pouvant être adapté, et optimisé plus aisément**. Le logiciel libre peut en effet être librement exécuté, copié, distribué, étudié, modifié et amélioré, sous réserve de respecter - dans les cas des licences copyleft - condition essentielle : la rediffusion du code source, modifié ou non, doit elle-même être libre et gratuite (en dehors des frais limités engendrés par la mise à disposition du code source), afin de protéger les travaux effectués par l'ensemble des programmeurs contre toute appropriation privative. Si cette condition n'est pas respectée, on considère qu'il y a contrefaçon du fait du non respect de la licence d'origine. C'est pourquoi certains considèrent que le logiciel libre a un effet contaminant sur les œuvres dérivées. Par exemple, l'article 2 de la licence GNU Limitée, Version 2.1, Février 1999 prévoit que si des applications d'un programme sont propriétaires, mais fonctionnent comme parties d'un ensemble cohérent dont le reste est libre, la licence libre s'appliquera à l'ensemble : il y aura donc diffusion.

Aujourd'hui, même s'il n'y a que peu de contentieux, les auteurs de logiciels libres n'hésitent plus à porter devant les tribunaux tout manquement aux clauses imposées dans leurs licences. Il devrait s'ensuivre un nombre important de procès, et il faudra regarder attentivement l'évolution de la jurisprudence sur ce point.

Il est donc essentiel de déterminer le cadre juridique commun des licences de logiciels libres et de mesurer l'impact que peut avoir le choix du logiciel libre.

7.1. Le cadre juridique des licences de logiciels libres

Il convient de rappeler certaines généralités sur le droit d'auteur et le logiciel libre et d'énoncer certaines spécificités. L'étude de la jurisprudence naissante permet d'avoir un aperçu sur ce que les licences de logiciels libres impliquent juridiquement.

Généralités : droit d'auteur et logiciel

Le droit d'auteur protège les « œuvres de l'esprit » au rang desquelles figurent les logiciels. Elles procèdent d'une activité créative et présentent un caractère d'originalité. Concrètement, cela implique que l'auteur de tout logiciel se voit reconnaître certains droits (droits moraux - notamment le droit à la paternité (art. L. 121-1 alinéa 1 du Code de la propriété intellectuelle) qui confère à l'auteur le droit au respect de son nom) et droits patrimoniaux) sur sa création dès lors que celle-ci présente un caractère d'originalité. Cette originalité est constituée en matière de logiciel par « la marque d'un apport intellectuel » (Babolat Maillot Witt / Pachot, C.Cass., Ass. Plén., 7 mars 1986) de l'auteur.

Le droit de reproduction (art. L. 122-3 du CPI – Code de la Propriété Intellectuelle) permet à l'auteur de détenir le monopole sur toutes les copies ou les fabrications à l'identique de son oeuvre ou de parties substantielles de son oeuvre.

Le droit de représentation (art. L. 122-2 du CPI) concerne le monopole de l'auteur s'agissant de la communication de son oeuvre au public.

C'est pourquoi, il est capital de procéder à une analyse minutieuse du contenu de la ou des licence(s) utilisée(s), avant de réaliser des développements informatiques à partir d'un logiciel libre.

Dans certains cas, les licences libres peuvent être contraignantes. Si l'utilisateur est libre d'utiliser le logiciel comme il l'entend, il peut aussi être contraint, dès lors qu'il modifie le logiciel, à respecter certaines obligations (distribution sous la même licence, impossibilité de réutiliser ce code modifié dans le contexte d'une application propriétaire, etc.).

Les obligations et les responsabilités attachées au logiciel libre sont directement issues de la licence choisie à l'origine par l'auteur du logiciel. Il existe, en effet, différentes licences qui se distinguent par leur caractère plus ou moins permissif.

Ainsi, le non-respect des obligations contenues dans la licence (par exemple obligation d'indiquer le copyright ou l'absence de redistribution du code source imposée par certaines licences libres) peut être sanctionné en tant que délit de contrefaçon prévu aux articles L. 335-2 et L. 335-3 du Code de la propriété intellectuelle instituant une peine maximum de 3 ans d'emprisonnement et de 300 000 euros d'amende (hors circonstances aggravantes).

Le brevet logiciel : un risque ou une opportunité ...

Les brevets logiciels sont reconnus par les tribunaux des États-Unis depuis les années 1980. Les acteurs industriels de l'informatique ont tous déposé des brevets sur des logiciels, comme ils étaient habitués à le faire sur du matériel. Dans ce contexte où tout développement de logiciel risque d'enfreindre d'innombrables brevets existants, certains acteurs du Logiciel Libre ont décidé de neutraliser les effets négatifs des brevets logiciels en créant l'Open Invention Network (OIN) et en nouant des alliances défensives. Par ce biais, ils tentent de déployer un bouclier dissuasif en achetant des brevets et en les mettant à disposition des développeurs libres.

En Europe, les brevets logiciels sont théoriquement interdits par le droit. Cependant l'Office européen des brevets en a déjà accordé, en interprétant abusivement les textes en la faveur du déposant. Dans ce contexte d'incertitude juridique, la meilleure défense à adopter pour les acteurs du Logiciel Libre est de s'unir pour peser politiquement sur des initiatives législatives mettant clairement fin aux pratiques contestables de l'Office européen des brevets. C'est notamment ce qui a permis, en 2005, de faire échec à une directive européenne qui devait consacrer la légalisation des brevets logiciels.

Or, même si la durée d'un brevet logiciel est limitée dans le temps, le fait de maîtriser la chaîne de création de nouveaux brevets pour de nouveaux logiciels assure une hégémonie perpétuelle des acteurs installés. Il sera alors très difficile, voire impossible, à de nouveaux entrants de développer et de commercialiser de nouveaux logiciels hors des écosystèmes fermés du club existant. Cette situation bloquerait l'innovation des PME européennes au profit d'une rente de situation perpétuelle pour des sociétés qui deviendraient les possédantes de l'économie de l'information. L'enjeu est du même ordre que celui qui a amené aux législations antitrust dans le cadre du capitalisme industriel.

Quelques spécificités des licences de « logiciels libres »

L'application du droit étranger

Les licences étant des contrats, elles tirent leur force de la loi. Aussi bien le droit d'auteur que le droit des contrats régissent donc les licences de logiciels libres. Cependant, la majorité de ces licences ne comportent aucune mention sur la loi applicable, ou elles se réfèrent à des concepts de droit étranger,

souvent anglo-saxon tel que le copyright, ou prévoient des clauses spécifiques comme par exemple ce que l'on retrouve aux articles 11 de la licence Netscape Public License V1, et Mozilla Public License :

This license shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provision.

Cette situation instaure un flou juridique, facteur d'insécurité pour le potentiel licencié qui ne sera pas en mesure de connaître les règles qui s'appliqueront en cas d'éventuels litiges.

Autre facteur d'insécurité juridique pour un licencié (utlisateur de la licence logiciel libre) français, les licences de logiciels libres sont pour la plupart rédigées en anglais, et les traductions, si elles existent, n'ont pas nécessairement un caractère officiel, en contravention avec l'article 2 de la loi Toubon du 4 août 1994 relative à l'emploi de la langue française qui énonce que :

« *Dans la désignation, l'offre, la présentation, le mode d'emploi ou d'utilisation, la description de l'étendue et des conditions de garantie d'un bien, d'un produit ou d'un service, ainsi que dans les factures et quittances, l'emploi de la langue française est obligatoire* ». Et l'article 5 de la même loi d'ajouter concernant les missions de service public : « *Quels qu'en soient l'objet et les formes, les contrats auxquels une personne morale de droit public ou une personne privée exécutant une mission de service public sont parties sont rédigés en langue française. Ils ne peuvent contenir ni expression ni terme étrangers lorsqu'il existe une expression ou un terme français de même sens approuvés dans les conditions prévues par les dispositions réglementaires relatives à l'enrichissement de la langue française* ».

En cas de non respect de ces dispositions, outre la nullité du contrat ainsi conclu, la sanction encourue est une contravention de quatrième classe, soit 750 euros par infraction constatée pour les personnes physiques et 3.750 euros pour les personnes morales (article 1er du décret du 3 mars 1995). Ainsi, aux termes de cette loi et d'une circulaire du 19 mars 1996 (J.O. du 20 mars 1996, p.4258), les distributeurs de logiciels libres seraient tenus d'employer la langue française, a minima lorsqu'ils contractent avec un consommateur français ou une administration française.

C'est pour pallier cela que certaines licences ont été rédigées par des organismes français, comme les licences CeCILL[12] ou les licences Art-Libre[13]. Elles font exception à la règle puisqu'elles comportent une clause relative à la loi applicable et désignent la loi française. Cela constitue une avancée en la matière et offre à un licencié français certaines garanties, notamment en matière de droits de propriété intellectuelle.

Au-delà du fait de savoir quelle loi est applicable, il est intéressant de s'interroger, dans l'hypothèse où la loi française est expressément désignée par le contrat, sur la conformité des dispositions contenues dans les licences libres au droit français. Cette question n'est pas aisée et soulève de nombreuses difficultés quant au formalisme et à la régularité des cessions des droits mais aussi quant aux garanties et responsabilités prévues dans le cadre de ces licences.

Compatibilité entre les licences

Pour distribuer un logiciel comportant des éléments de logiciels soumis à des licences libres différentes, il est nécessaire que ces différentes licences soient compatibles entre elles. Or, certaines licences libres ne sont pas compatibles entre elles en ce qu'elles imposent des obligations inconciliables.

[12] Les licences CeCILL sont disponibles sur le site : http://www.cecill.info

[13] Les licences Art-Libre sont disponibles sur le site : http://artlibre.org

Il apparait donc extrêmement important d'être vigilant lors de l'emploi de plusieurs briques logicielles Open Source et de vérifier la compatibilité des différentes licences entre elles.

Multilicensing

Le multilicensing permet à un auteur de logiciel libre de soumettre son œuvre à plusieurs licences libres. Cela permet de passer outre les difficultés liées aux incompatibilités de licences et le licencié choisira la licence proposée par l'auteur qui lui convient. Il se soumettra à cette licence et en respectera les termes.

Certains éditeurs ont fait le choix de distribuer leurs produits sous une double licence libre et propriétaire. Dans ce cas, l'intérêt pour l'éditeur est de fournir une version payante de son produit aux personnes qui ne voudraient pas être soumises aux obligations de la licence libre (par exemple MySQL AB distribué sous licence GPL (gratuite) sous le nom de MySQL server impose que chaque modification faite soit elle-même redistribuée sous cette licence (copyleft) ou sous licence commerciale (MySQL Enterprise).

L'adhésion à la licence de logiciel libre

Lorsqu'on télécharge une licence de logiciel libre, l'utilisateur se trouve en réalité dans le cadre d'un contrat d'adhésion, c'est-à-dire dans la même situation qu'en cas d'achat d'un logiciel propriétaire (par exemple, les « shrink wrap license » où l'acheteur du produit est lié par les clauses de la licence par le simple fait de déchirer l'emballage). Les clauses de la licence sont imposées par l'auteur et ne sont pas négociables. Pour illustration, il suffit de se reporter à l'article 5 de la licence Apache 2.0 de janvier 2004 (non traduite en français) qui stipule :

Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

Au final, soit le licencié accepte la licence et peut faire ce qui y est mentionné, soit il ne peut pas bénéficier de toutes les libertés inhérentes aux logiciels libres (modification et distribution). De plus, à partir du moment où l'acquisition du logiciel est gratuite (ce qui n'est pas l'hypothèse générale, mais la plus fréquente), la personne est moins vigilante. En effet, le plus souvent, elle ne prendra pas la peine de lire la licence et encore moins de conserver la version du contrat (électronique ou imprimée) applicable au moment du téléchargement.

Ainsi, le simple fait de télécharger ou d'utiliser le programme implique l'adhésion aux conditions sans en avoir pour autant pris connaissance. Cette façon de procéder est typiquement anglo-saxonne et ne se soucie en aucune façon des droits du licencié (consommateur, entreprise ou collectivité publique). A titre d'exemple, la licence Red Hat Linux dans sa version 1.1 indique dans son préambule :

Veuillez lire le présent contrat avant tout achat ou utilisation de produits ou prestations Red Hat. L'utilisation ou l'achat de produits ou prestations Red Hat valent consentement aux conditions générales de ce contrat. Si vous agissez au nom d'une personne morale, alors vous déclarez avoir autorité pour conclure ce contrat au nom de ladite personne morale. Si le client n'accepte pas les conditions générales du contrat, alors il ne doit pas acheter ni utiliser les produits et prestations Red Hat. Dans certains cas, il arrive cependant que l'acceptation de la licence soit liée au simple clic sur une case à cocher mais cela n'est pas le cas de toutes les licences de logiciels libres.

A cet égard, la licence GNU GPL présente une certaine particularité car il est précisé que l'utilisation du programme sous licence GPL n'implique pas nécessairement l'acceptation des clauses de la licence.

En revanche, dès lors que le licencié procède à des modifications ou à une redistribution du logiciel, il est supposé avoir implicitement accepté les conditions de la licence. En effet, l'article 5 de la GPL mentionne clairement :

Ne l'ayant pas signée, Vous n'êtes pas obligé d'accepter cette Licence. Cependant, rien d'autre ne Vous autorise à modifier ou distribuer le Programme ou quelques travaux dérivés : la loi l'interdit tant que Vous n'acceptez pas les termes de cette Licence. En conséquence, en modifiant ou en distribuant le Programme (ou tout travail basé sur lui), Vous acceptez implicitement tous les termes et conditions de la Licence.
Adaptation française non officielle de la GPL

L'article 9 de la LGPL (V.2.1 de février 1999) reprend une formulation quasiment identique à propos des bibliothèques logicielles.

Cette logique est particulièrement source d'insécurité juridique car finalement le licencié peut se trouver lié par les termes de la licence sans en avoir pris connaissance (utilisation libre et acceptation implicite en cas de modification ou de distribution du logiciel). En conséquence, une telle acceptation présente le risque de voir engagée une action en contrefaçon en cas de non respect des termes de la licence, l'autorisation initialement libertaire étant retirée de facto.

En outre, on observera que les termes contractuels de ces licences GPL interdisent expressément toute modification des termes mêmes de la licence, en disposant : « *La copie et la distribution de copies exactes de ce document sont autorisées, mais aucune modification n'est permise* ». De ce fait, la liberté initiale, ainsi que ses contraintes et obligations, se diffuse par le biais des adhésions successives.

Cette insécurité se traduit par un certain nombre de contentieux, voués à croître.

La jurisprudence en matière de logiciels libres

En l'état du droit et de la pratique des licences de logiciels libres, un constat s'impose : le logiciel libre a soulevé peu de contentieux.

L'affaire la plus médiatisée[14] a été celle qui a opposé SCO Group à IBM auquel il était reproché une « *concurrence illégale, rupture de contrat et appropriation abusive de secrets commerciaux* ». Selon SCO, le système d'exploitation GNU/Linux intégrerait une partie des sources UNIX dont elle a acquis les droits. A ce titre, SCO entend revendiquer les droits sur Linux. Un jugement, rendu le 10 août 2007 par la cour de district de l'Utah a considéré que c'était la société Novell et non la société SCO qui était titulaire des droits de la propriété intellectuelle sur le code source UNIX. Après cette décision, la société Novell a annoncé qu'elle ne souhaitait pas poursuivre l'action contre la société IBM. Il faut donc retenir que le donneur de licence doit veiller à ce que le code d'origine ne soit pas amélioré au moyen de codes sources protégés par un droit de propriété intellectuelle ou industrielle susceptibles de restreindre les libertés inscrites dans la licence.

En Allemagne, une décision du tribunal régional allemand de Munich, en date du 2 avril 2004, a condamné une société américaine qui n'avait pas respecté les termes de la licence GPL.

En France, les juges se sont prononcés dernièrement sur la validité de cette licence. Ainsi, la Cour d'appel de Paris, par un arrêt du 16 septembre 2009, a traité des effets juridiques d'une Licence portant sur un Logiciel Libre (voir Pascal Agosti, *Le Logiciel Libre en quête de reconnaissance juridique : les apports de l'arrêt de la Cour d'appel de Paris du 16 septembre 2009*, Journal du Net, 8 janvier

[14] Voir http://www.zdnet.fr/actualites/informatique/0,39040745,39205911,00.htm

2009).Jusqu'alors ils avaient statué sur l'effet contaminant du logiciel libre utilisé pour la création d'un programme :

« le programme en cause a la particularité de dépendre de la licence GNU qui permet une utilisation libre du logiciel mais requiert une licence si le travail basé sur le programme ne peut être identifié comme raisonnablement indépendant et doit être considéré comme dérivé du programme JATlite. »
TGI Paris, 3ème ch, 1ère sect., 28 mars 2007, Educaffix c/Cnrs

Ainsi, le programme développé était considéré comme un dérivé du logiciel libre.

L'arrêt de la Cour d'appel de Paris avait condamné une société informatique aux motifs qu'elle n'avait pas fourni à son client (l'AFPA) les sources d'un logiciel libre et d'avoir supprimé le texte de la licence GNU-GPL[15]. Suite à un appel d'offre important (plusieurs millions d'euros), l'AFPA a retenu la société EDU4. Lors de la recette de la phase 1 du marché, l'AFPA découvre que le logiciel VNC est un logiciel libre qui a été modifié par la société EDU4 et elle demande les sources à plusieurs reprises au prestataire, y compris par l'intermédiaire de la Free Software Fondation France (FSFFrance). Ce dernier refuse d'accéder à sa demande et lorsqu'il les fournit, ce ne sont pas celles qui correspondaient à la version livrée en 2001. La question du droit d'auteur n'a pas été abordée étant donné que l'auteur du logiciel n'était pas partie au litige. Cette décision met surtout en exergue l'inexécution du contrat justifiant sa résolution aux torts exclusifs du prestataire informatique. Dès lors on comprend l'importance que les décideurs doivent accorder quant au choix de leurs licences de logiciels libres et quant à la prévisibilité de leurs effets juridiques.

Enfin, il a été jugé par une Cour du district de San Francisco que le non respect d'une licence de logiciel libre n'entraîne pas seulement une responsabilité contractuelle se limitant à l'allocation de dommages et intérêts, mais aussi la résiliation automatique de la licence. Par conséquent, si le licencié ne respecte pas les obligations de la licence, il ne peut pas bénéficier des droits relatifs. Cependant, un jugement encore plus récent du 9 janvier 2009 est venu limiter les apports de cette décision.

2008 a vu également l'éclosion de procès majeurs : Cisco aux USA (la plainte de la Free Software Foundation contre CISCO date du 11 décembre 2008, elle rompt un dialogue et une collaboration entamée dès 2003 entre les deux structures), Free en France avec une assignation en date du 24 novembre 2008 : trois auteurs de logiciels libres reprochent à Free d'avoir distribués les logiciels « IPTABLES » et « BUSYBOX », tous deux sous GNU GPL sans respecter la licence. Free estime pour sa part qu'il n'y a pas de distribution (et donc pas de déclenchement de la licence). Il faut aussi noter la résolution à l'amiable de nombreux autres litiges (par exemple IchessU en Israël).

La responsabilité civile des SSLL

Un commettant (le dirigeant de la SSLL) est responsable des agissements de ses préposés (les développeurs) conformément à l'article 1384 al. 5 du Code civil. Dès lors, il faut être sûr que le recours à certains modules libres utilisés ne constitue pas une faute pouvant engager la responsabilité de la SSLL. Il est opportun de sensibiliser les développeurs à cette problématique et d'analyser en interne l'impact juridique des licences appliquées à certains modules libres.

7.2. Impacts liés au choix de « Licences de logiciels libres »

Quelque soit le type de licences, il existe toujours en droit français un droit moral d'auteur sur l'œuvre. Seulement, l'intensité de la protection de ce droit varie selon la licence choisie.

[15] Le texte de l'arrêt de la Cour d'appel de Paris est disponible aux adresses suivantes : www.caprioli-avocats.com et sur le site de la FSFFrance : http://fsffrance.org.

Contrairement à la tradition française conférant des garanties, ici, il y a une absence de garanties sur les applications développées sur le libre. Enfin, une contrainte supplémentaire, liée aux spécificités de distribution de logiciels de type copyleft, existe.

Avantages/inconvénients des licences libres selon le type

Simple utilisation

Il s'agit ici du cas dans lequel l'entreprise utilise un logiciel libre sans y apporter de modifications particulières.

Avantages :

- Le code source du logiciel est disponible, il est possible de l'auditer ou de le faire auditer, de l'adapter aux besoins de l'entreprise, le cas échéant (la modification pouvant entraîner certaines contraintes envisagées au point suivant).
- Le logiciel est généralement gratuit, le plus souvent le prix du logiciel sera constitué de la maintenance (assurée le cas échéant par l'éditeur ou par une société de service assurant l'intégration).
- L'entreprise n'est pas liée à un éditeur pour la maintenance ou l'intégration ou encore l'ajout de fonctionnalités, elle peut recourir à toute société de service de son choix par exemple.

Inconvénients :

- Les licences de logiciels libres écartent généralement la responsabilité des auteurs en cas de contrefaçon (reprise de code appartenant à un tiers).
- Il n'y a pas de garantie sur la qualité du logiciel comme c'est très souvent le cas pour les logiciels propriétaires.
- Il est parfois plus sécurisant de pouvoir se reposer sur un éditeur important afin de garantir le suivi dans le temps du logiciel.

Intégration d'un logiciel libre et modification

Est visé ici le cas dans lequel un progiciel sous licence libre est intégré, adapté et/ou paramétré en fonction des besoins de l'entreprise cliente.

La modification du logiciel touche le cas dans lequel l'entreprise modifie elle-même ou sous-traite à un prestataire informatique la modification d'un logiciel libre (ajout d'un module, ajout de fonction, correction de bug, interopérabilité avec la plateforme existant dans l'entreprise).

Avantages :

- La disponibilité du code source facilite grandement les possibilités de modifier le logiciel.
- L'utilisateur dispose (par la licence) du droit de réaliser ces modifications. Il n'est pas tenu au bon vouloir d'un éditeur pour réaliser ces modifications.
- Il sera possible de mettre en concurrence plusieurs prestataires pour réaliser ces modifications (celles-ci n'étant pas réservées à l'éditeur originel comme c'est souvent le cas des licences propriétaires).

Inconvénients :

- Lorsque la licence comporte un copyleft, les modifications ou les ajouts qui ne sont pas dissociables du logiciel libre original devront être redistribuées sous la même licence (libre). Cela ne vise toutefois que les licences disposant d'un copyleft fort (GNU GPL) et dès lors que la version modifiée du logiciel est redistribuée. Si la modification est faite pour un usage interne de l'entreprise et n'est donc pas redistribuée à des tiers, il n'y aura pas lieu d'appliquer les principes de mise à disposition du code source (rien n'empêchant l'entreprise de décider de le faire librement).
- A contrario, en cas de distribution, le fait de ne pas respecter les termes de la licence sera constitutif d'une contrefaçon.

Développement sous licence libre

Est envisagé dans ce cas, le cas où un logiciel est développé intégralement et soumis à une licence libre sans recours à une base de logiciels tiers.

Avantages :

- Possibilité de disposer d'une communauté d'utilisateurs permettant de tester et d'améliorer le logiciel.
- Implication de développeurs externes au projet original pouvant participer à son amélioration (bugs) ou au développement de nouvelles fonctionnalités sans coût supplémentaire.

Inconvénients :

- Le code source devant être mis à disposition du « public », il est difficile de commercialiser le logiciel en tant que tel. Le modèle économique reposera le plus souvent sur la prestation de service autour du logiciel (maintenance, développement spécifiques, intégration, mise à jour).
- Lorsque la licence ne comporte pas de copyleft, un tiers peut utiliser le logiciel, le modifier et redistribuer le logiciel modifié sous une autre licence (propriétaire par exemple dans le cadre d'une distribution commerciale).

Les garanties sur le logiciel

Un inconvénient de l'utilisation des logiciels libres réside dans l'absence de garantie sur le logiciel. En cela, les logiciels libres se distinguent nettement des logiciels propriétaires qui prévoient souvent des limitations de garantie mais contiennent dans la majorité des cas des garanties minimales (garantie de jouissance paisible ou d'éviction,..). Le licencié étant censé être à même de modifier et d'adapter le logiciel libre, on considère qu'il n'est pas nécessaire de lui octroyer des garanties supplémentaires.

Les articles 11 et 12 de la GNU GPL constituent un parfait exemple de l'absence de garantie attachée aux logiciels dits libres :

11. Comme la licence du programme est concédée à titre gratuit, aucune garantie ne s'applique au programme dans les limites autorisées par la loi applicable. Sauf mention contraire écrite, les titulaires du droit d'auteur et/ou les autres parties fournissent le programme « en l'état », sans aucune garantie de quelque nature que ce soit, expresse ou implicite, y compris, mais sans y être limité, les garanties implicites de commercialité et de la conformité à une utilisation particulière. Vous assumez la totalité des risques liés à la qualité et aux performances du programme. Si le programme se révélait

défectueux, le coût de l'entretien, des réparations ou des corrections nécessaires vous incombent intégralement.

12. En aucun cas, sauf lorsque la loi applicable ou une convention écrite l'exige, un titulaire de droit d'auteur quel qu'il soit, ou toute partie qui pourrait modifier et/ou redistribuer le programme comme permis ci-dessus, ne pourrait être tenu pour responsable à votre égard des dommages, incluant les dommages génériques spécifiques secondaires ou consécutifs, résultant de l'utilisation ou de l'incapacité d'utiliser le programme (y compris, mais sans y être limité, la perte de données, ou le fait que des données soient rendues imprécises, ou les pertes éprouvées par vous ou par des tiers, ou le fait que le programme échoue à interopérer avec un autre programme quel qu'il soit) même si le dit titulaire du droit d'auteur ou la partie concernée a été averti de l'éventualité de tels dommages.

Ces articles illustrent également le changement de modèle économique par rapport aux logiciels propriétaires dans la mesure où il est explicitement indiqué que le coût de la maintenance de ces logiciels incombe au licencié.

Le modèle français de logiciels libres à savoir CeCILL (version 2) reprend également cette logique d'absence de garantie dans son article 9 intitulé « Garantie » puisqu'il stipule :

9.1 Le Licencié reconnaît que l'état actuel des connaissances scientifiques et techniques au moment de la mise en circulation du Logiciel ne permet pas d'en tester et d'en vérifier toutes les utilisations ni de détecter l'existence d'éventuels défauts. L'attention du Licencié a été attirée sur ce point sur les risques associés au chargement, à l'utilisation, la modification et/ou au développement et à la reproduction du Logiciel qui sont réservés à des utilisateurs avertis.

Il relève de la responsabilité du Licencié de contrôler, par tous moyens, l'adéquation du produit à ses besoins, son bon fonctionnement et de s'assurer qu'il ne causera pas de dommages aux personnes et aux biens.

9.2 Le Concédant déclare de bonne foi être en droit de concéder l'ensemble des droits attachés au Logiciel (comprenant notamment les droits visés à l'article 5).

9.3 Le Licencié reconnaît que le Logiciel est fourni « en l'état » par le Concédant sans autre garantie, expresse ou tacite, que celle prévue à l'article 9.2 et notamment sans aucune garantie sur sa valeur commerciale, son caractère sécurisé, innovant ou pertinent.

En particulier, le Concédant ne garantit pas que le Logiciel est exempt d'erreur, qu'il fonctionnera sans interruption, qu'il sera compatible avec l'équipement du Licencié et sa configuration logicielle ni qu'il remplira les besoins du Licencié.

Outre l'absence de garantie sur le logiciel lui-même, la majorité des licences de logiciels libres est également exempte de garantie de jouissance paisible du logiciel, contrairement aux conditions d'utilisation de logiciels propriétaires. A titre d'illustration, la licence CeCiLL V.2 précise en son article 9.4 :

Le Concédant ne garantit pas, de manière expresse ou tacite, que le Logiciel ne porte pas atteinte à un quelconque droit de propriété intellectuelle d'un tiers portant sur un brevet, un logiciel ou sur tout autre droit de propriété. Ainsi, le Concédant exclut toute garantie au profit du Licencié contre les actions en contrefaçon qui pourraient être diligentées au titre de l'utilisation, de la modification, et de la redistribution du Logiciel.

Néanmoins, si de telles actions sont exercées contre le Licencié, le Concédant lui apportera son aide technique et juridique pour sa défense. Cette aide technique et juridique est déterminée au cas par cas

entre le Concédant concerné et le Licencié dans le cadre d'un protocole d'accord. Le Concédant dégage toute responsabilité quant à l'utilisation de la dénomination du Logiciel par le Licencié. Aucune garantie n'est apportée quant à l'existence de droits antérieurs sur le nom du Logiciel et sur l'existence d'une marque.

De ce point de vue, il faut reconnaître que le licencié est moins bien protégé que lorsqu'il contracte une licence propriétaire avec un éditeur de logiciel. Il doit donc prendre à sa charge tous les risques inhérents à l'utilisation du logiciel. Cela relève d'un choix du licencié.

A cet égard, les clauses limitatives ou exclusives de responsabilité sont en principe valables en droit français. Mais il est des cas où ces clauses seront déclarées nulles, inefficaces, inopposables ou réputées non écrites. Il en sera ainsi en vertu de la législation sur les clauses abusives qui protège le consommateur. Le caractère abusif de ce type de clause à l'égard du consommateur ne fait aucun doute dans la mesure où il est stipulé que le logiciel libre est fourni « *sans aucune garantie de quelque nature que ce soit* ». Toutefois, il apparaît nécessaire de préciser que ces clauses exonératoires de responsabilité pourraient être considérées comme nulles sur le fondement des grands principes du droit Civil et du droit de la Consommation. En conséquence, cette clause risque d'être écartée, même dans les relations entre professionnels, en cas de manquement à une obligation essentielle du contrat, ayant pour effet de contredire la portée de l'engagement pris. Pour finir, les clauses limitatives ou exclusives de responsabilité seront déclarées nulles en cas de faute lourde ou de dol[16] de celui qui les invoque. Tous ces éléments sont une fois de plus constitutifs d'une certaine incertitude juridique tant pour le concédant que pour le licencié. Il en est de même pour la distribution des logiciels de type copyleft.

La distribution des logiciels de type copyleft

Les licences de type copyleft sont des licences qui permettent l'utilisation, la modification et même la redistribution du logiciel, à condition cependant que le logiciel modifié soit redistribué sous la même licence que le logiciel initial. La licence GNU GPL en est une parfaite illustration. Ainsi aux termes de son article 2, il est mentionné :

« Vous pouvez modifier votre copie ou des copies du Programme ou n'importe quelle partie de celui-ci, créant ainsi un ouvrage fondé sur le Programme, et copier et distribuer de telles modifications ou ouvrage selon les termes de l'Article 1 ci-dessus, à condition de vous conformer également à chacune des obligations suivantes :

- *Vous devez munir les fichiers modifiés d'avis bien visibles stipulants que vous avez modifié ces fichiers, ainsi que la date de chaque modification;*
- *Vous devez prendre les dispositions nécessaires pour que tout ouvrage que vous distribuez ou publiez, et qui, en totalité ou en partie, contient ou est fondé sur le Programme - ou une partie quelconque de ce dernier - soit concédé comme un tout, à titre gratuit, à n'importe quel tiers, au titre des conditions de la présente Licence.*
- *Si le programme modifié lit habituellement des instructions de façon interactive lorsqu'on l'exécute, vous devez, quand il commence son exécution pour ladite utilisation interactive de la manière la plus usuelle, faire en sorte qu'il imprime ou affiche une annonce comprenant un avis de droit d'auteur ad hoc, et un avis stipulant qu'il n'y a pas de garantie (ou bien indiquant que c'est vous qui fournissez la garantie), et que les utilisateurs peuvent redistribuer le programme en respectant les présentes obligations, et expliquant à l'utilisateur comment voir une copie de la présente Licence.*

[16] Définition issue de Wikipédia: «un dol est une manoeuvre d'un cocontractant dans le but de tromper son partenaire et provoquer chez lui une erreur. Le dol est, avec l'erreur et la violence, l'un des trois vices du consentement. Il est sanctionné par la nullité du contrat.». Plus de détails : http://fr.wikipedia.org/wiki/Dol_(droit)

(Exception : si le Programme est lui-même interactif mais n'imprime pas habituellement une telle annonce, votre ouvrage fondé sur le Programme n'est pas obligé d'imprimer une annonce).

Ces obligations s'appliquent à l'ouvrage modifié pris comme un tout.

Si des éléments identifiables de cet ouvrage ne sont pas fondés sur le Programme et peuvent raisonnablement être considérés comme des ouvrages indépendants distincts en eux mêmes, alors la présente Licence et ses conditions ne s'appliquent pas à ces éléments lorsque vous les distribuez en tant qu'ouvrages distincts.

Mais lorsque vous distribuez ces mêmes éléments comme partie d'un tout, lequel constitue un ouvrage fondé sur le Programme, la distribution de ce tout doit être soumise aux conditions de la présente Licence, et les autorisations qu'elle octroie aux autres concessionnaires s'étendent à l'ensemble de l'ouvrage et par conséquent à chaque et toute partie indifféremment de qui l'a écrite.

Par conséquent, l'objet du présent article n'est pas de revendiquer des droits ou de contester vos droits sur un ouvrage entièrement écrit par vous; son objet est plutôt d'exercer le droit de contrôler la distribution d'ouvrages dérivés ou d'ouvrages collectifs fondés sur le Programme.

De plus, la simple proximité du Programme avec un autre ouvrage qui n'est pas fondé sur le Programme (ou un ouvrage fondé sur le Programme) sur une partition d'un espace de stockage ou un support de distribution ne place pas cet autre ouvrage dans le champ d'application de la présente Licence ».

De plus, l'article 4 dispose expressément :

Vous ne pouvez copier, modifier, concéder en sous-licence, ou distribuer le Programme, sauf tel qu'expressément prévu par la présente Licence. Toute tentative de copier, modifier, concéder en sous-licence, ou distribuer le Programme d'une autre manière est réputée non valable, et met immédiatement fin à vos droits au titre de la présente Licence. Toutefois, les tiers ayant reçu de vous des copies, ou des droits, au titre de la présente Licence ne verront pas leurs autorisations résiliées aussi longtemps que lesdits tiers se conforment pleinement à elle.

A la lecture de ces dispositions contractuelles, plusieurs risques juridiques liés à l'utilisation de ces licences peuvent être identifiés : la résolution de la licence (article 4 de la licence GPL) et l'action en contrefaçon. En effet, selon certains auteurs :

la condition de la renonciation est que l'usager qui adaptera le programme laisse à son tour librement les tiers se servir de l'oeuvre dérivée tirée de celui-ci. C'est une sorte de condition résolutoire à l'acte juridique, en général énoncée dans la licence gratuite. Si l'adaptateur tente de s'approprier en exclusivité son amélioration, la licence sera résolue de plein droit à ses torts, sans préjudice d'un éventuel grief de contrefaçon formulé par l'auteur originaire. Tel est l'esprit du logiciel libre.

Ainsi l'utilisation des logiciels libres peut présenter des avantages mais il ne faut pas pour autant occulter les incertitudes juridiques qui en découlent. Aussi, une lecture attentive des licences et un archivage de celles-ci au moment de leur téléchargement sont fortement recommandés.

Contrairement à une idée reçue, les logiciels libres ne doivent pas être opposés aux logiciels propriétaires, car les deux modèles sont complémentaires et peuvent être combinés. Aucun modèle de développement n'est adapté à tous les utilisateurs, à tous les clients. Mais l'analyse des risques et des conséquences juridiques qui est souvent faite dans le cadre des logiciels propriétaires, ne peut être occultée lors de l'acquisition d'un logiciel libre.

Les principales licences Open Source et leurs caractéristiques :

	Auteur	Parution (dernière version)	Droit Moral	Compatibilité avec GNU GPL	Diffusion
Apache	Apache Foundation	Janvier 2004	Flou juridique	Compatible GNU GPL v3 uniquement	Permissive
CeCILL	CEA, CNRS, INRIA	21 mai 2005	Conforme au droit français	Compatible GNU GPL	Copyleft
CeCILL B	CEA, CNRS, INRIA		Conforme au droit français	Non compatible avec GNU GPL Compatible BSD	Copyleft
CeCILL C	CEA, CNRS, INRIA		Conforme au droit français	Compatible GNU GPL	Copyleft
GNU GPL	Richard Stallman	29 juin 2007	Flou juridique	Voir liste sur le site de la FSF	Copyleft
EUPL	Commission européenne	09 janv. 2007	Respecte les usages contractuels de l'ensemble des Etats membres de l'Union	Compatible GNU GPL v2, mais pas encore compatible avec la v3	Copyleft

8. LES LEVIERS DU DÉVELOPPEMENT OPEN SOURCE

8.1. Pour une théorie du développement libre

Lorsque Frederic Brooks était en charge, dans les années 60, du développement de l'OS/360 à IBM, il rencontra tellement de difficultés qu'il mit les bases du Génie Logiciel en place et les consigna, en 1974, dans un livre devenu célèbre : The Mythical Man-Month.

Se mirent alors en place des méthodes de développement pour le logiciel.

Lorsque Internet commença à se développer, apparurent des techniques anarchiques de développement logiciel dont le support fut le réseau naissant. Ce phénomène a été analysé et décrit par Eric Stephen Raymond dans son essai : The Cathedral and the Bazaar[17]. En simplifiant, on pourrait dire qu'un logiciel commercial est plutôt produit à partir d'un processus de type Cathédrale tandis qu'un logiciel libre est plutôt issu d'un processus de type Bazar, mais il y a plus intéressant.

Au départ, la majorité des logiciels développés à travers le réseau étaient des "briques", de petits outils, qui pouvaient être développés par un nombre restreint de programmeurs en interaction relativement directe, comme dans un Bazar. L'apport du réseau était que l'on n'était plus obligés de se rencontrer pour travailler ensemble mais il n'y avait pas d'effet de levier. La taille des équipes et des logiciels produits de la sorte restait la même que dans le monde "réel".

Pourtant, un des plus beaux succès de ces techniques est, sans doute, le développement de GNU/Linux. C'est un système complet, dont la complexité du développement, à la fois par la taille et le nombre des équipes, les interactions entre les différents modules, demande qu'une gestion de projet rigoureuse soit mise en place. Comme pour la construction d'une cathédrale, un architecte et une équipe de contremaîtres pour assurer que ses ordres sont respectés semblent indispensables. Malgré tout, le modèle de développement en réseau utilisé par les développeurs de GNU/Linux s'est avéré efficace aussi sur un projet de cette envergure.

8.2. C'est un problème de gestion de projet !

Processus Cathédrale

La bonne gestion d'un projet suppose que :

- les objectifs sont clairement définis,
- le déroulement est soigneusement planifié,
- l'exécution est soigneusement contrôlée (coûts, délais, qualité),
- les résultats sont validés,
- le projet est clôturé.

Certains types de projets ne peuvent être correctement conduits que de cette façon. Ce processus de conduite a donné naissance au métier de Chef de Projet reconnu par une certification professionnelle, de nombreuses formations privées ou universitaires existent.

Un tel projet qui échoue laisse des ruines dont le déblaiement va constituer un nouveau projet. L'échec n'est pas une option.

[17] Raymond, Eric, The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, O'Reilly, 2001 aussi disponible en ligne à l'adresse : http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/ (Déjà cité page 16)

L'impact de l'ouvrage The mythical Man-Month[18] a conduit à considérer le développement logiciel sur le même plan que la construction d'un ouvrage d'art et de considérer que la bonne gestion de projet s'applique également au développement logiciel.

Processus Bazar

Le développement logiciel est un projet original qui ne manipule pas de matière. Cet aspect immatériel a permis d'envisager d'autres méthodes de conduite de projet.

Un projet immatériel qui échoue ne laisse pas de ruines; on va simplement l'oublier. L'échec peut être une option.

Dans ce contexte, une gestion de projet de type collaborative est possible :

- faire un programme qui marche,
- le publier au plus vite pour le soumettre à la critique constructive du monde,
- donner aux utilisateurs la possibilité effective de corriger ses défauts ou de l'améliorer,
- collecter et publier les modifications au plus vite.

Le plus surprenant est que cela fonctionne plutôt bien, au moins dans certains cas. Mais pour cela des conditions un peu particulières sont à satisfaire :

- le programme doit présenter un intérêt suffisamment général pour que beaucoup d'utilisateurs se sentent concernés,
- il doit être suffisamment utile pour que l'utilisateur qui détecte un défaut (un manque) ait intérêt à le modifier (compléter) lui-même ... puis à communiquer la modification (le complément) en retour, l'utilisateur devient alors un contributeur,
- la publication des modifications doit être (très) rapide pour ne pas décourager les bonnes volontés.

Les contributions peuvent être très variées selon les capacités de chacun :

- correction d'un dysfonctionnement logiciel,
- participation au visuel en modifiant des images,
- participation à l'internationalisation en contribuant à la traduction,
- rédaction d'une notice, d'un tutoriel,
- présentation de conférences,
- rédaction d'articles,
- suggestion de nouvelles fonctionnalités,
- etc.

Ces contributions peuvent être de nature tellement variées qu'il est assez facile de s'écarter du modèle de pur Bazar.

Un risque pour l'auteur d'un logiciel libre qui entend en retirer un certain bénéfice est le départ d'un fork. Ainsi, si un logiciel libre est, a priori, fourni en l'état, sans engagement d'aucune sorte, négliger le support aux utilisateurs-contributeurs peut en faire perdre le bénéfice à son auteur.

Le fork n'est pas forcément le diable ; c'est aussi une garantie de pérénité supérieure à la durée de vie de l'auteur.

[18] Brooks, Fred, The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley, 1975

8.3. Développer un logiciel libre, qu'est-ce que ça change ?

Techniquement (presque) rien.

Développer un logiciel avec l'intention d'en libérer le code ne change pas grand chose par rapport au développement d'un logiciel commercial.

Après tout, développer un logiciel ne dépend pas de la façon de le distribuer. Par contre, ce qui va changer, c'est la façon de le maintenir (corrections et évolutions) :

- un logiciel propriétaire est maintenu par son éditeur,
- un logiciel libre est maintenu par sa communauté de contributeurs.

8.4. Aider les contributeurs devient vite indispensable

Il est vrai que le partage de code est vite une façon naturelle de travailler lorsqu'on a l'intention de partager son propre code. Ce partage de code est une ancienne tradition que certaines habitudes assez récentes (ainsi que l'attitude de certains éditeurs de logiciels) ont pu faire oublier pendant quelques (dizaines) d'années.

Cependant, s'il est possible de libérer un logiciel après coup, le développement d'un logiciel destiné à être libéré se passera beaucoup mieux si ce fait a été intégré dès le départ. En particulier, toute l'attention nécessaire aura été portée sur le choix des licences à utiliser.

C'est dans cette optique que ce chapitre est rédigé.

On considère également que le logiciel a été développé libre ou libéré dans l'optique de créer une communauté d'utilisateurs et de contributeurs.

8.5. Distribuer le logiciel, ses Sources et sa Licence

La seule contrainte est la distribution des sources à la demande.

Différentes formes de distribution existent :

- Sous la forme d'une machine virtuelle
- Sous forme binaire directement installable
- Sous forme compilable (avec les scripts de compilation)
- etc.

Si l'envoi d'un CD par La Poste répond tout à fait au cahier des charges, le téléchargement peut être préférable.

Dans le cas d'un logiciel libre payant, il est assez fréquent qu'une version limitée gratuite soit téléchargeable.



8.6. Assurer le support

Assurer le support revient très vite à mettre en place un système de gestion de la communauté :

- FAQ
- Liste de diffusion
- Forum de discussion
- Espace de téléchargement
- Espace de téléversement
- Traqueur de défauts
- Espace d'internationalisation
- etc.

À cela peut se rajouter un espace de développement :

- Gestion des versions
- Compilation
- etc.

8.7. Documentation

Il est indispensable de fournir avec tout logiciel une documentation adéquate.

8.8. Remontée des défauts de fonctionnement

C'est la remontée rapide des défauts de fonctionnement bloquants qui va permettre d'obtenir un logiciel dont la qualité augmente suffisamment vite pour continuer à motiver ses utilisateurs. Cette remontée peut être effectuée à travers un simple forum pour les projets de petite envergure, par contre, un système de ticketing est vite indispensable pour les projets de grande ampleur.

Si la remontée des défauts est importante la signalisation de leur correction l'est également.

Il fut un temps où les contributeurs (informaticiens) remontaient les défauts après les avoir corrigés. Aujourd'hui la notion de contributeur s'est largement élargie et contribuer n'est pas forcément réservé aux programmeurs. Il est donc devenu naturel de demander la correction d'un défaut.

8.9. Demandes de nouvelles fonctionnalités

Il est des demandes de nouvelles fonctionnalités comme des défauts. En fait, il ne serait pas étrange de considérer qu'une fonctionnalité qui manque est un défaut vis à vis de l'usage.



8.10. A considérer également

- La gestion des mises à jour
- La gestion des contributeurs
- La gestion des utilisateurs

8.11. Les Forges

Les Forges logicielles sont des outils collaboratifs couramment utilisés afin de centraliser et d'organiser le travail de plusieurs dizaines voire centaines de contributeurs, développeurs, traducteurs…

Ceux sont les outils incontournables de l'organisation et de la gestion d'une communauté.

Elles permettent de mettre en musique tous les processus énumérés ci-dessus.

CHECK LIST (EN GUISE DE CONCLUSION… PROVISOIRE…)

Cette check list est la liste de quelques questions simples que toute personne devrait se poser si elle envisage d'intégrer des briques Open Source dans son développement, d'utiliser des briques Open Source dans son système d'information ou de publier son logiciel sous licence Open Source.

Nous espérons que la lecture de ce livre blanc vous aura donné un nouvel éclairage vous permettant de répondre plus facilement et efficacement à ces quelques questions :

- ✓ Qu'est ce que je produis et où se trouve la réelle valeur ajoutée de mon produit ?
- ✓ Quelles sont les barrières à l'entrée sur le marché ciblé ?
- ✓ Y'a-t-il un intérêt à distribuer ce logiciel sous licence Open Source ?
- ✓ Comment optimiser le levier marketing procuré par le choix d'une licence Open Source ?
- ✓ Quel modèle économique adopter pour maximiser mes chances de succès ?
- ✓ La licence que j'ai choisie est-elle en adéquation avec mon modèle économique ?
- ✓ Ai-je besoin ou mon client a t-il besoin de garanties spécifiques ?
- ✓ Ai-je bien analysé les risques et conséquences juridiques de mon choix de Licence ?
- ✓ Quelles sont les exigences légales à l'utilisation de la brique Open Source choisie (diffusion, reversement de code…) ?
- ✓ J'utilise des composants Open Source, ai-je le droit de les livrer avec mon logiciel sous licence propriétaire ?
- ✓ Ai-je intérêt à créer une communauté ?
- ✓ Comment créer une communauté afin de bénéficier de ce levier ?
- ✓ Suis-je prêt à l'animer ? Avec quels moyens humains, financiers et techniques ? De quelle manière ?
- ✓ Ai-je pensé à la manière de fédérer utilisateurs et contributeurs ?
- ✓ Ai-je pensé à la manière de faire vivre et pérenniser cette communauté ?

ANNEXES

Terminologie

Dans le cadre de ce livre blanc les termes suivants ont été utilisés avec cette signification :

Code objet	Un programme compréhensible par l'ordinateur dans un format binaire exécutable. (Définition : http://www.marche-public.fr).
Code source	Ensemble d'instructions écrites dans un langage de programmation informatique de haut niveau, c'est-à-dire humainement compréhensible, permettant d'obtenir un programme pour un ordinateur, est mis à la disposition du public.
Contamination	Référence aux cas dans lesquels un logiciel propriétaire doit être distribué avec une licence libre car il a été « contaminé » par le logiciel libre avec lequel il fonctionne.
Standard ouvert	Tout protocole de communication, d'interconnexion ou d'échange et tout format de données interopérable et dont les spécifications techniques sont publiques et sans restriction d'accès ni de mise en œuvre (art. 4 de la LCEN[19]).
Interopérabilité	Capacité que possède un produit ou un système, dont les interfaces sont intégralement connues, à fonctionner avec d'autres produits ou systèmes existants ou futurs et ce sans restriction d'accès ou de mise en œuvre.
Licences de type copyleft	Licences permettant l'utilisation, la modification et même la redistribution du logiciel, à condition cependant que le logiciel modifié soit redistribué sous la même licence que le logiciel initial. En d'autres termes, l'utilisateur qui modifie un logiciel sous licence libre avec copyleft, ne disposera pas de droits patrimoniaux sur l'œuvre modifiée. Il sera obligé de la distribuer avec une licence libre lui aussi. Il devra obligatoirement fournir le code objet et le code source.
Logiciel libre	Logiciel dont la licence (dite libre) qui l'accompagne accorde à l'utilisateur les libertés suivantes : • liberté d'exécution, sans restriction de temps, de lieu de motif, de personne, etc., • liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins, • liberté de redistribuer des copies, • liberté d'améliorer et de diffuser les améliorations. *(définition Free Software Foundation).*

[19] La Loi pour la Confiance en l'Economie Numérique (LCEN) est accesible à l'adresse suivante: http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000000801164

Logiciel open source	Logiciel dont la licence doit répondre à des critères pour être qualifiée en tant que telle : • la libre redistribution du logiciel, • le code source doit être fourni et accessible, • l’intégrité du code doit être préservée, • pas de discrimination entre les personnes ou les groupes, • pas de discrimination entre les domaines d’application, • la licence est indépendante des autres contrats, • la licence ne doit pas être propre à un produit, • la licence a un logiciel ne doit pas s’étendre à un autre, • la licence doit être neutre technologiquement. Il existe peu de distinctions entre « logiciels libres » et « Open Source » car en pratique les licences définies comme libres par la *Free Software Foundation* (FSF) et l'*Open Source Initiative* (OSI), cf. licences libres, sont identiques à quelques cas anecdotiques près.
Logiciel propriétaire	Logiciel dont seuls les codes objet sont mis à disposition des utilisateurs et dont l’utilisation est encadrée par une licence restrictive.
Multilicensing	Technique permettant à un auteur de logiciel libre de soumettre son œuvre à plusieurs licences libres ou propriétaires.
SSLL / SS2L	Société de Services en Logiciels Libres.

Historique

La version originale d'un chapitre de ce Livre Blanc a été publiée par l'association April http://www.april.org en décembre 2007. Elle est disponible selon les termes de la GNU Free Documentation Licence version 1.2 ou ultérieure publiée par la Free Software Foundation. Les versions PDF ou Open Document Format sont disponibles sur le site de l'April.

- version 1.0 : décembre 2007

Les membres contributeurs du groupe de travail entreprises :

- Kathryn Fairlamb, Adacore
- Véronique Torner & Philippe Montargès, AlterWay
- Jérôme Dumonteil & Yvon Rastetter, Ars Aperta
- Luis Belmar Letelier, Itaapy
- Nicolas Chauvat, Logilab

sous la coordination de Jean-Noël de Galzain, Wallix.

La version actuelle, complétée, modifiée et enrichie de ce livre blanc a été publiée par l'association Telecom Valley http://www.telecom-valley.fr en décembre 2010. Elle est disponible selon les termes de la GNU Free Documentation Licence version 1.2 ou ultérieure publiée par la Free Software Foundation. La version PDF est disponible sur le site de Telecom Valley.

- version 1.0.- Décembre 2010

Les membres contributeurs du groupe de travail Open Source de Telecom Valley :

- Pascal Agosti, Cabinet Caprioli & Associés
- Marc Augier, SKEMA
- Jean Demartini, DEMTECH
- Christophe Deneux, Petals ESB
- Juliette Mallez, @CTIS-Ingénierie

sous la coordination de Pascal Flamand, Janua.

Licence GNU Free Documentation Licence

GNU Free Documentation licence

Version 1.2,

November 2002

0. PREAMBLE

The purpose of this licence is to make a manual, textbook, or other functional and useful document « free » in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially.

Secondarily, this licence preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This licence is a kind of « copyleft », which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public licence, which is a copyleft licence designed for free software.

We have designed this licence in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this licence is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this licence principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This licence applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this licence. Such a notice grants a world-wide, royalty-free licence, unlimited in duration, to use that work under the conditions stated herein. The « Document », below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as « you ». You accept the licence if you copy, modify or distribute the work in a way requiring permission under copyright law.

A « Modified Version » of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A « Secondary Section » is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The « Invariant Sections » are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this licence. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The « Cover Texts » are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this licence. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A « Transparent » copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent.

An image format is not Transparent if used for any substantial amount of text. A copy that is not « Transparent » is called « Opaque ».

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The « Title Page » means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this licence requires to appear in the title page. For works in formats which do not have any title page as such, « Title Page » means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section « Entitled XYZ » means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as « Acknowledgements », « Dedications », « Endorsements », or « History ».) To « Preserve the Title » of such a section when you modify the Document means that it remains a section « Entitled XYZ » according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this licence applies to the Document. These Warranty Disclaimers are considered to be included by reference in this licence, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this licence.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this licence, the copyright notices, and the licence notice saying this licence applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this licence. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's licence notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this licence, with the Modified Version filling the rôle of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a licence notice giving the public permission to use the Modified Version under the terms of this licence, in the form shown in the Addendum below.

G. Preserve in that licence notice the full lists of Invariant Sections and required Cover Texts given in the Document's licence notice.

H. Include an unaltered copy of this licence.

I. Preserve the section Entitled « History », Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled « History » in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the « History » section.

You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled « Acknowledgements » or « Dedications », Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled « Endorsements ». Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled « Endorsements » or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's licence notice.

These titles must be distinct from any other section titles.

You may add a section Entitled « Endorsements », provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this licence give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this licence, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its licence notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this licence, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of Invariant Sections in the licence notice of the combined work.

In the combination, you must combine any sections Entitled « History » in the various original documents, forming one section Entitled « History »; likewise combine any sections Entitled « Acknowledgements », and any sections Entitled « Dedications ». You must delete all sections Entitled « Endorsements ».

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this licence, and replace the individual copies of this licence in the various documents with a single copy that is included in the collection, provided that you follow the rules of this licence for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this licence, provided you insert a copy of this licence into the extracted document, and follow this licence in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an « aggregate » if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit.

When the Document is included in an aggregate, this licence does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

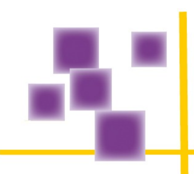
If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form.

Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4.

Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this li-





cence, and all the licence notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this licence and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this licence or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled « Acknowledgements », « Dedications », or « History », the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this licence. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this licence. However, parties who have received copies, or rights, from you under this licence will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENCE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation licence from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the licence is given a distinguishing version number.

If the Document specifies that a particular numbered version of this licence « or any later version » applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this licence, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this licence for your documents

To use this licence in a document you have written, include a copy of the licence in the document and put the following copyright and licence notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation licence, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the licence is included in the section entitled « GNU Free Documentation License ».

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the « with...Texts. » line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software licence, such as the GNU General Public licence, to permit their use in free software.



Remerciements

L'association April tient à remercier tous les membres de l'association ayant aidé à la relecture de ce Livre Blanc :

Jean-Christophe Becquet, Frédéric Couchet, Pierre Cros, Nicolas Dumoulin, Gilles Gravier, Benjamin Jean, Yann Kerbiriou, Arnaud Laprévote, Christophe Le Bars, Agnès Le Béon, Véronique Gouilly-Frossard, Eric Mahé, Eva Mathieu, Charles-Henri Schulz, Gérald Sedrati-Dinet, Benoît Sibaud.

L'April remercie également toutes les entreprises ayant répondu à l'enquête qui a permis d'établir les cartographies.

L'association Telecom Valley tient à remercier tous les membres ainsi que l'équipe opérationnelle de l'association ayant aidé à la création et à la relecture de ce Livre Blanc.

Telecom Valley remercie également tout particulièrement l'April[20] dont le document sur les modèles économiques du logiciel libre inspire très largement notre chapitre correspondant.



[20] www.april.org